Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia

FAULT DETECTION, DIAGNOSIS AND FAULT TOLERANCE APPROACHES IN DYNAMIC SYSTEMS BASED ON BLACK-BOX MODELS

LUIS FILIPE FIGUEIRA BRITO PALMA

Dissertação submetida à Universidade Nova de Lisboa
para obtenção do grau de Doutor em Engenharia Electrotécnica –
Especialidade de Controlo

Portugal, Lisboa – 2007

Fault Detection, Diagnosis and Fault Tolerance Approaches in Dynamic Systems based on Black-Box Models

Copyright © Luís Filipe Figueira Brito Palma, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

The Faculdade de Ciências e Tecnologia and the Universidade Nova de Lisboa have a right, perpetual and without geographical limits, of filing and publishing this dissertation through printed examples reproduced in paper or in the digital form, or for any other known way or that might be invented, and of spreading it through scientific repositories and of admitting its copy and distribution with education or investigation objectives, without commercial intents, as credit is given to the author and publisher.

Advisor ("Orientador"): Prof. Doutor Fernando José Almeida Vieira do Coito

Co-Advisor ("Co-Orientador"): Prof. Doutor Rui Alexandre Nunes Neves da Silva

Jury ("Júri"):

President ("Presidente"):

Prof. Doutora Zulema Paula do Perpétuo Socorro Lopes Pereira

Examiner Committee ("Comité de Arguência"):

Prof. Doutor Luís Miguel de Mendonça Rato

Prof. Doutor António Eduardo de Barros Ruano

Evaluation Committee ("Comité de Avaliação - Vogais"):

Prof. Doutor Hermínio Duarte-Ramos

Prof. Doutor António Dourado Pereira Correia

Prof. Doutor Fernando José Almeida Vieira do Coito

Prof. Doutor Rui Alexandre Nunes Neves da Silva

Prof. Doutor Luís Miguel de Mendonça Rato

Prof. Doutor António Eduardo de Barros Ruano

Acknowledgments

It is doubtful that anyone arrives at the end of their dissertation without a good deal of help along the way; this author is certainly no exception. High on the list of those to thank is my advisor, Professor Fernando Coito. Next on the list is my co-advisor Professor Rui Neves da Silva. It was a great pleasure to work with them. Their tough scientific questions helped me think very carefully about what I was doing and has forced me to clarify my own thinking as well as being a great help in achieving the research objectives. I also thank them for having believed in me, for the scientific and technical support, and for the constant motivation.

I would also like to thank all the members of the Section of Systemic Engineering (Decision and Control - SDC) of the Departamento de Engenharia Electrotécnica (DEE) of Faculdade de Ciências e Tecnologia (FCT) of Universidade Nova de Lisboa (UNL). From the beginning of the creation of the DEE, Professor Duarte-Ramos has always helped me think as a systemic engineer, and transmitted to me the importance of the systems and signals, and their role in all research areas. My office colleague, Professor Paulo Gil, deserves special thanks, for the motivation transmitted and for the scientific and technical support given in our discussions. To the other colleagues of the SDC, I offer my thanks for supporting my exemption from teaching service for three years.

A special thanks also goes to the Faculdade de Ciências e Tecnologia (FCT-UNL) and to Fundação para a Ciência e Tecnologia (FCT-MCT), for supporting my exemption from teaching service for three years, for the technical support, and for their financial support. Thanks also to Uninova for their financial support.

I would also like to thank Professors R. Isermann, P. Frank, R. Patton, B. Wise, M. Harkat, U. Kruger, and J. Lopes for the papers that they sent me. For their technical support in the laboratory, I would like to thank Manuel Quintela and Luís Leandro, and my final project course student, Rui Almeida. I would also like to thank all my teachers, and to other people who have helped me to achieve the end of this project.

The work presented in this dissertation was done in the Section of Systemic Engineering of the DEE-FCT-UNL, in Monte de Caparica, Portugal, and also in collaboration with the Control of Industrial Plant (CIP) research group of Uninova.

The work developed along the last years, and resumed in this dissertation, is dedicated to my wife Paula, and my daughters Carolina and Sofia. I would like to thank my family for the support, and for tolerating my faults.

Abstract

In this dissertation new contributions to the research area of fault detection and diagnosis in dynamic systems are presented. The main research effort has been done on the development of new on-line model-based fault detection and diagnosis (FDD) approaches based on black-box models (linear ARX models, and neural nonlinear ARX models). From a theoretical point of view a white-box model is more desirable to perform the FDD tasks, but in most cases it is very hard, or even impossible, to obtain. When the systems are complex, or difficult to model, modelling based on black-box models is usually a good and often the only alternative. The performance of the system identification methods plays a crucial role in the FDD methods proposed.

Great research efforts have been made on the development of linear and nonlinear FDD approaches to detect and diagnose multiplicative (parametric) faults, since most of the past research work has been done focused on additive faults on sensors and actuators.

The main pre-requisites for the FDD methods developed are: a) the on-line application in a real-time environment for systems under closed-loop control; b) the algorithms must be implemented in discrete time, and the plants are systems in continuous time; c) a two or three dimensional space for visualization and interpretation of the fault symptoms. An engineering and pragmatic view of FDD approaches has been followed, and some new theoretical contributions are presented in this dissertation.

The fault tolerance problem and the fault tolerant control (FTC) have been investigated, and some ideas of the new FDD approaches have been incorporated in the FTC context.

One of the main ideas underlying the research done in this work is to detect and diagnose faults occurring in continuous time systems via the analysis of the effect on the parameters of the discrete time black-box ARX models or associated features. In the FDD methods proposed, models for nominal operation and models for each faulty situation are constructed in off-line operation, and used a posteriori in on-line operation.

The state of the art and some background concepts used for the research come from many scientific areas. The main concepts related to data mining, multivariate statistics (principal component analysis, PCA), linear and nonlinear dynamic systems, black-box models, system identification, fault detection and diagnosis (FDD), pattern recognition and discriminant analysis, and fault tolerant control (FTC), are briefly described. A sliding window version of the principal components regression algorithm, termed SW-PCR, is proposed for parameter

estimation. The sliding window parameter estimation algorithms are most appropriate for fault detection and diagnosis than the recursive algorithms.

For linear SISO systems, a new fault detection and diagnosis approach based on dynamic features (static gain and bandwidth) of ARX models is proposed, using a pattern classification approach based on neural nonlinear discriminant analysis (NNLDA). A new approach for fault detection (FDE) is proposed based on the application of the PCA method to the parameter space of ARX models; this allows a dimensional reduction, and the definition of thresholds based on multivariate statistics. This FDE method has been combined with a fault diagnosis (FDG) method based on an influence matrix (IMX). This combined FDD method (PCA & IMX) is suitable to deal with SISO or MIMO linear systems.

Most of the research on the fault detection and diagnosis area has been done for linear systems. Few investigations exist in the FDD approaches for nonlinear systems. In this work, two new nonlinear approaches to FDD are proposed that are appropriate to SISO or MISO systems. A new architecture for a neural recurrent output predictor (NROP) is proposed, incorporating an embedded neural parallel model, an external feedback and an adjustable gain (design parameter). A new fault detection and diagnosis (FDD) approach for nonlinear systems is proposed based on a bank of neural recurrent output predictors (NROPs). Each neural NROP predictor is tuned to a specific fault. Also, a new FDD method based on the application of neural nonlinear PCA to ARX model parameters is proposed, combined with a pattern classification approach based on neural nonlinear discriminant analysis.

In order to evaluate the performance of the proposed FDD methodologies, many experiments have been done using simulation models and a real setup. All the algorithms have been developed in discrete time, except the process models. The process models considered for the validation and tests of the FDD approaches are: a) a first order linear SISO system; b) a second order SISO model of a DC motor; c) a MIMO system model, the three-tank benchmark. A real nonlinear DC motor setup has also been used. A fault tolerant control (FTC) approach has been proposed to solve the typical reconfiguration problem formulated for the three-tank benchmark. This FTC approach incorporates the FDD method based on a bank of NROP predictors, and on an adaptive optimal linear quadratic Gaussian controller.

Keywords: fault detection, fault diagnosis, fault tolerance, fault tolerant control, black-box models, dynamic systems.

Résumé

Dans cette dissertation, de nouvelles contributions sont présentées dans le domaine de la recherche en rapport à la détection et diagnostic de fautes (pannes, défauts) en systèmes dynamiques. Le principal effort de recherche a été fait sur le développement des nouvelles méthodologies en ligne pour détection et diagnostic de fautes (FDD), utilisant des modèles boîte noire (ARX linéaires et ARX neuronaux non linéaires). D'un point de vue théorique, un modèle boîte blanc est plus approprié pour exécuter les tâches FDD, mais dans la plupart des cas, il est très difficile, ou même impossible, à obtenir. Lorsque les systèmes sont complexes, ou difficiles à modéliser, la modélisation basée sur des modèles boîte noire est généralement bonne et, souvent, la seule alternative. La performance des méthodes d'identification de systèmes joue un rôle crucial dans les méthodologies FDD proposées.

Grands efforts de recherche ont été réalisés sur le développement de méthodologies FDD linéaires et non linéaires pour détecter et diagnostiquer fautes multiplicatifs (paramétriques), puisque dans le passé la plupart des travaux de recherche a été fait sur fautes additifs en capteurs et en actionneurs. Les principales pré-requis pour les méthodes FDD sont: a) la application en temps réel pour les systèmes de contrôle en boucle fermée; b) les algorithmes doivent être mises en oeuvre en temps discret et les installations sont des systèmes en temps continu; c) deux ou trois dimensions spatiales pour la visualisation et l'interprétation des symptômes des fautes. Il a été suivie une perspective de ingénierie et pragmatique des approches FDD, et quelques nouveaux contributions théoriques sont présentés.

La tolérance aux fautes et le contrôle tolérant aux fautes (FTC) ont été investigués et quelques idées des nouvelles approches FDD ont été incorporées dans le contexte FTC.

L'une des principales idées qui est sous-jacent à la recherche effectuée est détecter et diagnostiquer les fautes survenants dans les systèmes en temps continu d'après l'analyse des effets sur les paramètres des modèles ARX ou caractéristiques associées. En ce qui concerne les méthodes FDD proposées, de modèles de fonctionnement nominale et de modèles pour chaque situation de faute sont construits hors ligne et utilisés en ligne a posteriori.

L'état de l'art et certains concepts utilisés pour la recherche provienne de nombreux domaines scientifiques. Les principales concepts liées à l'extraction de données, statistique multivariée (analyse en composantes principales, PCA), systèmes dynamiques linéaires et non linéaires, modèles boîte noire, identification de systèmes, détection et diagnostic de fautes (FDD), la reconnaissance des patrons (formes) et analyse discriminante, et le contrôle tolérant aux

fautes, sont brièvement décrits. Une version de fenêtre coulissante de l'algorithme de régression en composantes principales est proposée pour l'estimation des paramètres, soyant plus approprié pour la détection et le diagnostic de fautes que les algorithmes récursifs.

Pour les systèmes SISO linéaires, une nouvelle méthodologie FDD basée sur caractéristiques dynamiques (gain statique et bande passante) de modèles ARX est proposée, utilisant une approche de classification de patrons basée sur l'analyse discriminante neuronaux non linéaire (NNLDA). Il a été proposé une nouvelle approche de détection des fautes (FDE) basée sur l'application de la méthode PCA à l'espace de paramètres des modèles ARX; cela permet une réduction des dimensions et la définition de seuils fondé sur la statistique multivariée. Cette méthode basée en PCA a été combinée avec une méthode de diagnostic des fautes (FDG) fondée sur une matrice d'influence, et il est approprié pour systèmes linéaires.

La plupart des recherches dans le domaine de la détection et diagnostic de fautes a été faite pour les systèmes linéaires. Il existe peu d'études rapportés avec les approches FDD en systèmes non linéaires. Dans ce travail, deux nouvelles approches non linéaires sont proposées pour FDD. Une nouvelle architecture pour un prédicteur neuronal récurrent (NROP) est proposée, intégrant un modèle parallèle, une retroaction externe et un gain réglable. Une nouvelle méthodologie FDD est proposée basée sur un ensemble de prédicteurs NROPs. Chaque prédicteur neuronal NROP est syntonisé à une faute. Aussi, une nouvelle méthode FDD est proposée, fondée sur l'application de PCA neuronal non linéaire aux paramètres du modèle ARX, combinée avec un schéma de classification de patrons.

Afin d'évaluer la performance des methodologies FDD, de nombreuses expériences ont été effectuées en utilisant des modèles de simulation et une installation réelle. Tous les algorithmes ont été mis au point en temps discret, à l'exception des modèles de processus. Les modèles des processus considérés pour la validation et les tests sont: a) un système linéaire SISO de premier ordre; b) un système linéaire SISO de deuxième ordre (modèle d'un moteur à courant continu); c) un modèle d'un système MIMO (le banc d'essais à trois réservoirs COSY). Un moteur non linéaire DC a également été utilisé. Une approche de contrôle tolérant aux fautes a été proposé pour résoudre le problème typique de reconfiguration formulé pour le banc d'essais à trois réservoirs COSY. Cette approche intègre la méthode FDD fondée sur un ensemble de prédicteurs NROPs et un contrôleur optimal adaptative quadratique gaussien.

Mots clés: détection de fautes (pannes, défauts), diagnostic de fautes, tolérance aux fautes, contrôle tolérant aux fautes, modèles boîte noire, systèmes dynamiques.

Resumo

Nesta dissertação são apresentadas novas contribuições para a área de investigação em detecção e diagnóstico de falhas em sistemas dinâmicos. O principal esforço de investigação foi realizado no desenvolvimento de novas metodologias de detecção e diagnóstico de falhas (FDD) baseadas em modelos caixa-preta (modelos lineares ARX, e modelos neuronais NARX não-lineares) com aplicação em linha. De um ponto de vista teórico é preferível o uso de modelos caixa-branca para realizar as tarefas de FDD, mas na maioria das situações é muito difícil ou mesmo impossível obtê-los. Quando os sistemas são complexos, ou difíceis de modelizar, a modelização baseada em modelos caixa-preta é usualmente uma boa e frequentemente a única alternativa. O desempenho dos métodos de identificação de sistemas tem um papel crucial nos métodos de FDD propostos.

Elevados esforços de investigação foram feitos no desenvolvimento de metodologias de FDD para detecção de falhas multiplicativas (paramétricas), dado que a maioria do trabalho de investigação tem se centrado nas falhas aditivas em sensores e actuadores.

Os principais pré-requisitos dos métodos de FDD desenvolvidos são: a) a aplicação em linha em ambientes de tempo-real para sistemas controlados em anel fechado; b) os algoritmos devem ser implementados em tempo discreto, e as instalações são sistemas em tempo contínuo; c) uma visualização e interpretação a duas ou três dimensões dos sintomas das falhas. Uma perspectiva de engenharia e pragmática das metodologias de FDD foi seguida, e algumas contribuições teóricas são apresentadas nesta dissertação.

O problema da tolerância a falhas em sistemas dinâmicos e o controlo tolerante a falhas foi investigado, e algumas ideias das novas metodologias de FDD foram incorporadas no contexto do FTC.

Uma das ideias principais subjacente à investigação realizada neste trabalho é a detecção e diagnóstico de falhas que ocorrem em processos em tempo contínuo, através da análise do efeito nos parâmetros de modelos discretos caixa-preta do tipo ARX ou características associadas. Nos métodos de FDD propostos são construídos em diferido modelos para funcionamento nominal e modelos para funcionamento em várias situações de falhas, e usados à posteriori em funcionamento em linha.

O estado da arte e alguns conceitos utilizados na investigação provêm de várias áreas científicas. Os conceitos principais, brevemente descritos, estão relacionados com análise exploratória de dados, estatística multi-variada (análise em componentes principais, PCA), sistemas dinâmicos lineares e não lineares, modelos caixa-preta, identificação de sistemas,

detecção e diagnóstico de falhas (FDD), reconhecimento de padrões e análise discriminante, e controlo tolerante a falhas (FTC). Uma versão do tipo janela deslizante do algoritmo de regressão por componentes principais, denominado SW-PCR, é proposto para estimação de parâmetros. Os algoritmos de estimação de parâmetros do tipo janela deslizante são mais apropriados para detecção e diagnóstico de falhas do que os algoritmos recursivos.

Para sistemas lineares SISO, uma nova metodologia de detecção e diagnóstico de falhas é proposta sendo baseada em características dinâmicas (ganho estático e largura de banda) de modelos ARX, utilizando um método de classificação de padrões baseado em análise discriminante neuronal não linear (NNLDA). A análise discriminante NNLDA permite a definição de fronteiras de decisão necessárias para detecção e isolamento de falhas, sendo mais eficiente que as técnicas geométricas. Uma nova metodologia para detecção de falhas (FDE) é proposta tendo como base a aplicação da análise PCA ao espaço de parâmetros de modelos ARX; tal permite uma redução de dimensão, e a definição de limiares baseados em estatística multi-variada. Este método de FDE é combinado com um método de diagnóstico de falhas (FDG) baseado numa matriz de influência. Este método combinado para FDD (PCA & IMX) é apropriado para ser aplicado em sistemas lineares SISO ou MIMO.

A maioria da investigação na área de detecção e diagnóstico de falhas tem sido realizada para sistemas lineares. Poucas investigações existem relacionadas com as metodologias FDD para sistemas não lineares. Neste trabalho, duas novas metodologias de FDD para sistemas não lineares são propostas, sendo apropriadas para sistemas SISO ou MIMO. Uma nova arquitectura para um preditor neuronal recorrente da saída (NROP) é proposta, incorporando um modelo neuronal paralelo embutido, um anel de retroacção e um ganho de ajuste (parâmetro de projecto). Um novo método de FDD para sistemas não lineares é proposto tendo como base um banco de preditores neuronais recorrentes (NROPs). Cada preditor neuronal NROP é sintonizado para uma falha específica. Também é proposto um novo método de FDD baseado na aplicação de PCA neuronal não linear aos parâmetros de modelos ARX de forma a lidar com sistemas não lineares, combinado com um método de classificação de padrões baseado em análise discriminante neuronal não linear.

De forma a avaliar o desempenho das metodologias de FDD propostas, várias experiências foram realizados em modelos de simulação e num equipamento real. Todos os algoritmos foram desenvolvidos em tempo discreto. Os modelos dos processos considerados para a validação e testes das metodologias de FDD são todos em tempo contínuo: a) um sistema

linear SISO de primeira ordem; b) um modelo de segunda ordem de um motor de corrente contínua; c) um modelo de sistema MIMO, o sistema de três tanques "three-tank benchmark". Também foi usado um motor real de corrente contínua, com características não lineares, para teste das metodologias de FDD. Uma metodologia de controlo tolerante a falhas (FTC) foi proposta para resolver o problema típico de reconfiguração para o sistema de três tanques "three-tank benchmark". Esta metodologia de controlo tolerante a falhas incorpora a metodologia de detecção e diagnóstico de falhas baseada num banco de preditores neuronais recorrentes (NROPs) proposta neste trabalho.

Palavras-chave: detecção de falhas, diagnóstico de falhas, tolerância a falhas, controlo tolerante a falhas, modelos caixa-preta, sistemas dinâmicos.

Notation and Symbols

Notation.

AEM: abnormal event management

ARX: auto-regressive linear model with exogenous input

DAQ: data acquisition system

DMI: data mining

DPCA: dynamic PCA

EKF: extended Kalman filter

FDE: fault detection

FF: feed-forward

FIO: fault isolation

FDA: Fisher Discriminant Analysis

FDI: fault detection and isolation

FID: fault identification (or analysis)

FDG: fault diagnosis

FDD: fault detection and diagnosis

FTC: fault tolerant control

IIR: infinite impulse response (filter)

IMX: influence matrix

MISO, MIMO: multi-input single-output system; multi-input multi-output system

MLP: multi-layer perceptron

MNT: monitoring

MSE: mean of squared errors

MVS: multivariate statistics

NARX: nonlinear auto-regressive model with exogenous input

NNLDA: neural nonlinear discriminant analysis

NLPCA: nonlinear PCA

NN: neural networks, also mentioned Artificial NN (ANN)

PCA: linear principal components analysis

PC: principal components

PCR: principal components regression

PID: proportional-integral-derivative

RLS: recursive least squares

SISO: single-input single-output system

SPE: square of prediction error

SPV: supervision

SPTC: Safeprocess Technical Committee

SS: state-space (model)

SSE: sum of squared errors

SVD: singular value decomposition

Symbols and Operators.

Variables and scalars are represented by small letters in italic, (ex., $x(t)$, α, …). Matrices are represented by capital letters in bold, (ex., $\mathbf{X}$, $\mathbf{X}(:,:,k)$, $\mathbf{\Sigma}$, …). Vectors are represented in small letters, in bold, (ex., $\mathbf{x}(:,k)$**, y,** …).

t	Continuous time variable
k	Discrete time variable
$x(t)$	Continuous time signal at instant time t
$x(k)$	Discrete time signal at discrete time k
$\mathbf{X}(:,j)$, $\mathbf{X}(j,:)$	$\mathbf{X}(:,j)$, j^{th} column of matrix $\mathbf{X}$; $\mathbf{X}(j,:)$, j^{th} line of matrix $\mathbf{X}$
$\mathbf{X}(:,:,k)$	Matrix $\mathbf{X}$ at time instant k
$\mathbf{y}(k)$	Vector $\mathbf{y}$ of n samples at time k; $\mathbf{y}(k) = \mathbf{y}(1{:}n,k)$
$TZ\{x(k)\}$	Z-transform of signal $x(k)$; $X(z) = TZ\{x(k)\}$
$TZ^{-1}\{X(z)\}$	Inverse Z-transform of $X(z)$; $x(k) = TZ^{-1}\{X(z)\}$
q, q^{-1}	Forward and backward shift operators: $q\,y(k) = y(k+1)$; $q^{-1}\,y(k) = y(k-1)$
T_S	Sampling period
$\Re^{n \times m}$	Euclidian "n x m" dimensional space
$H_{lp}(z, \lambda)$	Low pass IIR filter
$N(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2

Models

$G_0(s)$	Transfer function of a plant in continuous time
$G_0(z)$	Transfer function of a plant in discrete time
θ	Vector used to parameterize models; dimension = d
M; M^*	Model structure; set of models

$M(\theta)$	Particular model corresponding to the parameter value θ
M_P	Process model
M_X	Auto-regressive linear model with exogenous input (ARX model); ARX(n_a, n_b, n_d); n_a, n_b, n_d are the model orders, and the pure time delay
$M_{yu}(\theta)$, $M_{yr}(\theta)$	Input-output ARX model relating the output y with the input u, and reference-output ARX model relating the output y with the reference signal r
M_{XN}	Nonlinear ARX model
M_{NN}	Neural model
M_{NP}	Neural predictor
M_{NO}	Neural observer
M_{NNLDA}	Neural model for nonlinear discriminant analysis
M_F; M^*_F	Fault model; set of fault models

PCA

$\mathbf{X} \in \Re^{n \times m}$	Data matrix for nominal operation (without faults)
n, m	Number of lines and columns of data matrix $\mathbf{X}$
M_{PCA}	Linear PCA model
M_{NLPCA}	Nonlinear PCA model
$\hat{\mathbf{X}}$	Estimation of data matrix $\mathbf{X}$ from PCA model
a	Number of principal components retained by the PCA model (dimension of the scores subspace)
$\mathbf{V} \in \Re^{m \times m}$	Matrix containing the loading vectors

Table of Contents

List of Figures

List of Tables

1 Introduction

If I were waiting for perfection, I never would have written this book (Tai T'ung, History of Chinese Writing, century XIII).

1.1 Motivations

The operation of industrial technical processes requires increasingly advanced supervision, fault detection and diagnosis (FDD) approaches, and fault tolerant control (FTC) methods to increase reliability, safety and economy. One of the main goals of the fault tolerant systems is to guarantee that faults do not cause dangerous failures, and even human fatalities. In many publications, the Abnormal Situation Management (ASM) Consortium estimates that the costs associated with some types of failures in technical processes are many million of euros per year, and that they can be drastically reduced if fault tolerant control systems are implemented. The ASM is a consortium of companies and universities that are concerned about the negative effects of industrial plant incidents. The FTC systems can also improve the maintenance policies, and increase the reliability of the overall plant.

The research area related to fault detection and diagnosis, and to fault tolerant control, is a fascinating and complex area of research, mainly due to the interactions of different research areas like control, data mining, statistics, soft computing, etc.

For the author of this dissertation, these are the main motivations to work in this research area.

1.2 Main Goals and Contributions

In the past, the main research efforts in the fault detection and diagnosis (FDD), and in fault tolerant control (FTC) areas, have been mostly done based on analytical approaches using white-box models, and considering mainly additive faults on sensors and actuators.

The main goals of this dissertation are to focus on the detection and diagnosis of multiplicative (parametric) faults on process components, using FDD methods based on

black-box models (linear ARX models and nonlinear ARX neural models), and on (linear and nonlinear) principal component analysis, that do not require a precise mathematical model of the process. The faults can occur on the process, on the actuators, on the sensors, or on the controller.

Mainly, two types of black-box models are considered in this work: the autoregressive ARX linear model, and the nonlinear ARX neural model.

FDD approaches based on classical PCA and neural nonlinear PCA are proposed.

The developed FDD and FTC methods were implemented in order to satisfy on-line and real-time specifications, and to work in closed-loop operation. These specifications are important if one of the goals is to apply the FDD/FTC methods in real plants, like industry plants.

The experience acquired along this work has shown that, in most of the situations, a combination of different FDD methodologies is necessary to implement practical and robust approaches.

The basic philosophy underlying this dissertation is that a pragmatic approach is the road to success. One of the most important lessons to be learned from the numerous automatic control applications developed over the past half century is that simple solutions actually solve most problems quite well.

Here, special efforts were made in the direction of the development of FDD methods with a geometrical interpretation in two or three dimensions.

The main contributions given in this dissertation are described next.

A new fault detection and diagnosis method based on dynamic features (static gain and bandwidth) of ARX models is proposed to deal with linear systems, where a pattern recognition method based on neural nonlinear discriminant analysis (neural NNLDA) is used.

A new approach for fault detection in linear systems is proposed. This new statistical fault detection approach is based on the application of principal component analysis (PCA) to the parameters space of ARX models. This fault detection approach based on PCA is combined with a fault diagnosis approach based on an influence matrix (IMX) method.

A new architecture for a neural output predictor (NROP) is proposed, incorporating an embedded parallel model, an external feedback, and an adjustable gain (design parameter). A new fault detection and diagnosis (FDD) approach for nonlinear systems is proposed based on a bank of neural recurrent output predictors (NROPs). Each neural predictor (NROP) is tuned to a particular fault.

A new FDD method based on the application of neural nonlinear PCA to ARX model parameters is proposed to deal with nonlinear systems. This method is combined with a pattern recognition method based on neural nonlinear discriminant analysis (neural NNLDA).
The fault tolerance problem in dynamic systems and the fault tolerant control (FTC) have been investigated, and some ideas of the new FDD approaches have been incorporated in the FTC context. A fault tolerant control (FTC) approach has been proposed to solve the typical reconfiguration problem formulated for the three-tank benchmark. This fault tolerant control approach incorporates the fault detection and diagnosis method based on a bank of neural recurrent output predictors (NROPs) proposed in this work, and use an adaptive optimal linear quadratic Gaussian controller (LQGC).

1.3 Dissertation Layout

This dissertation is organized as follows.
In Chapter 1, the motivations, the main goals and contributions, and the dissertation layout is presented.
Chapter 2 contains the state-of-the-art and the background used for the research that comes from several scientific areas. The main concepts used in this dissertation are included and discussed.
In Chapter 3, two new contributions for FDD in linear systems are described. The first contribution is the FDD method based on dynamic features of ARX models, and on neural nonlinear discriminant analysis (neural NNLDA). The other contribution is the statistical fault detection method based on Principal Component Analysis; this method is combined with the fault diagnosis method based on the influence matrix method.
Chapter 4 is dedicated to fault detection and diagnosis approaches for nonlinear systems, where a brief review of the main approaches is described. The new neural recurrent output predictor (NROP) is presented. The proposed FDD approach based on a bank of neural recurrent output predictors (NROPs) is described. A new FDD method based on the application of neural nonlinear PCA to ARX model parameters is proposed to deal with nonlinear systems, combined with a neural nonlinear discriminant analysis (neural NNLDA) for pattern classification.
The main experimental results obtained using the proposed fault detection and diagnosis approaches applied to process models and plants are presented in Chapter 5. Results for a

fault tolerant control (FTC) approach to solve the typical reconfiguration problem formulated for the three-tank benchmark are also shown.

Finally, the conclusions and the future work appear in Chapter 6.

2 State of the Art

A perfectly accurate and complete mathematical model of a physical system is never available (L. B. Palma).

2.1 Introduction

The state-of-the-art is presented, and the main concepts used in this work are included and discussed in this chapter. The ground work for this research comes from many scientific areas. Concepts related to data mining, multivariate statistics, linear and nonlinear dynamic systems, black-box models, system identification, fault detection and diagnosis (FDD) and fault tolerant control (FTC) are briefly reviewed.

In the scientific and technical communities the terminology is not consensual. Some definitions are presented here in order to clarify any terms which may be misunderstood. It is beyond the scope of this document to give a complete treatment of all of these fields. Enough background information is included, so that the reader may follow the developments and extensions proposed in the subsequent chapters.

It is the natural option to represent the signals in discrete time, primarily since observed data are always collected by sampling and processed by a personal computer.

2.2 Data Mining

Data mining is a science related to knowledge discovery (important information) from data (Berry & Linoff, 1997). Some common definitions are: a) "the science of extracting useful information from large data sets or databases"; b) "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data". To do this, data mining uses mainly computational techniques from statistics, pattern recognition, and machine learning. The concept of data mining is also known as Exploratory Data Analysis.

The recent industrial plants are becoming more and more complex, requiring real-time supervision, and generating high quantities of process data. In the last two decades, mainly due to the use of high efficient dedicated computers and advanced instrumentation, the

monitoring and supervision of industrial systems allows a high quality performance, and a wide understanding of the plant behaviour. The high quantity of available data requires an efficient extraction of information from this data. Based on the information, it is necessary to extract knowledge from the process, in order to perform many important tasks like monitoring, fault detection and diagnosis, control and supervision. An emergent paradigm in many sciences is: "from data to information and knowledge". According to Ackoff (1989) the content of the human mind can be classified into five categories: data, information, knowledge, understanding and wisdom.

A great challenge in many areas, and in particularly in the area of supervision and process control, is to deal with high quantities of available data captured from sample-data systems with many experimental variables (eventually correlated). In this context, the increase of information in quantity is mainly due to the increase in the sampling frequency, and the increase of information in quality is mainly due to the increase of monitored variables (Lopes, 2001). The development of intelligent knowledge-based methods that learn from data, like neural networks and fuzzy systems, allow engineers to deal with high quantities of data. The data mining concepts have been applied in many fields: fault detection and diagnosis, econometrics, chemometrics, data clustering, pattern classification, and multivariate statistics.

The reduction of dimensionality is fundamental when the goal is to discover patterns that are not clear in multivariate data. The most popular data-driven method for reduction of dimensionality is the Principal Component Analysis (Jackson, 2003).

2.3 Process Monitoring and Multivariate Statistics

2.3.1 Introduction

Multivariate statistics, or multivariate statistical analysis, is a statistical science that deals with observation and analysis of more than one statistical variable at a time.

The effectiveness of the data-driven measures depends on the characterization of the process data variations. There are two types of variations for process data (Ogunnaike & Ray, 1994; Chiang, et al., 2001): common cause and special cause. The common cause variations are those due to random noise (e.g., associated with sensor readings), whereas special cause variations account for all the other data variations. Since variations in the process data are unavoidable, statistical theory plays an important role in most process monitoring schemes.

The application of statistical theory to process monitoring relies on the assumption that the characteristics of the data variations are relatively unchanged, unless a fault occurs in the system. If a Fault is defined as "an abnormal process condition" (Chiang, et al., 2001), this is a reasonable assumption. It implies that the properties of the data variations, such as the mean and variance, are repeatable for the same operating conditions; the repeatability of the statistical properties allows the establishing of thresholds for certain measures.

Multivariate statistical techniques have been applied in many scientific and technological areas (Piovoso & Kosanovich, 1996; Chiang, et al., 2001). The majority of applications of some techniques of multivariate statistics have been in chemical process industries, power plants, and nuclear plants.

Within the scope of the present work, the main focus is on the areas of process monitoring and fault detection and diagnosis in dynamic systems modeled by black-box models.

This section will describe briefly the main techniques of multivariate statistics used: Principal Component Analysis (PCA), and Principal Component Regression (PCR). There is special emphasis on how to use statistical methods for process monitoring, in particular methods using the multivariate *T2* statistics and the *Q* statistics computed based on the PCA approach. PCR is used for estimation of ARX model parameters.

2.3.2 Data Pre-Processing

To extract the information in the data relevant to process monitoring, it's often necessary to pre-process the data in the training set. The training set contains off-line data available for analysis prior to the on-line implementation of the process monitoring schemes, and is used to develop the measures representing the nominal operations (without faults) and the faulty situations. The pre-processing procedures consist of three tasks (Chiang, et al., 2001): removing variables, auto-scaling, and removing outliers.

Inappropriate variables in the training set that have no information relevant to process monitoring should be removed before further analysis.

The second task, auto-scaling (also termed standardization), is usually necessary to avoid particular variables dominating the process monitoring method, especially in those methods based on dimensionality reduction techniques, such as Principal Component Analysis (PCA). Auto-scaling standardizes the process variables, ensuring that for each variable an equal weight is given. Each variable is subtracted by its sample mean (μ), so to capture the variation of the data from the mean. Then each variable of the mean-centered data is divided by its standard deviation (σ). These two steps scale each variable to mean zero, and unit variance.

When auto-scaling is applied to new process data, the mean to be subtracted and the standard deviation to be divided are taken from the training set.

Finally, the outliers should be removed. Outliers are isolated measurement values that are erroneous. These values may significantly influence the estimation of statistical parameters (or other parameters) related to a given measure.

2.3.3 Normal Distribution

The normal distribution, also called Gaussian distribution, is a probability distribution of great importance in many fields such as statistical data analysis, fault detection and diagnosis, medicine, psychology, biology, financial studies, etc. The normal distribution is probably the most important distribution in both the theory and application of statistics. A definition of its probability distribution function (PDF) is given next.

Definition 2.1. If y is a normal random variable, in continuous time, then the probability distribution function of y is given by (Montgomery, 1991):

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2} \quad -\infty < y < +\infty \qquad \text{Eq. 2.1}$$

The parameters of the normal distribution are the mean μ ($-\infty < \mu < +\infty$) and the variance $\sigma^2 > 0$. A typical notation, $y \sim N(\mu, \sigma^2)$, is used to imply that y is normally distributed.

■

The standard univariate normal distribution is the normal distribution with a mean of zero ($\mu = 0$) and a variance of one ($\sigma^2 = 1$). The normality assumption is at the core of a majority of standard statistical procedures. The graph of its probability distribution function (PDF) resembles a symmetric bell-shaped curve. The best visual way to evaluate how far the data are from Gaussian is to look at a graph and see if the distribution deviates from a bell-shaped normal distribution.

Definition 2.2. If y is a normal random variable, then the cumulative normal distribution function is defined as the probability that the normal variable y is less than or equal to some value b (Montgomery, 1991):

$$\Pr\{y \le b\} = F(b) = \int_{-\infty}^{b} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2} dy \qquad \text{Eq. 2.2}$$

■

Several methods exist to test normality, for example the Kolmogorov-Smirnov test, the normal probability (rankit) plot, the Lilliefors test, the Shapiro-Wilk test, the D'Agostino-Pearson omnibus test, etc.

The normal distribution is often assumed as the appropriate probability model for a random variable. In many instances, it is difficult to check the validity of this assumption; however, the central limit theorem is often a justification of approximate normality (Montgomery, 1991). The so-called central limit theorem (CLT) formulated by H. Cramér around 1937 appears in many different forms. All forms have in common the purpose of stating conditions under which the sum of several random variables, regardless of the distributions of the individual variables, may be approximated by a normal random variable. The central limit theorem can be stated as follows: if the sample size is large, the theoretical sampling distribution of the mean can be approximated closely with a normal distribution. Another formulation of this theorem is given below (Conover, 1999).

Theorem 1. Central Limit Theorem.
Let Σ_n be the sum of n random variables $x_1, x_2, \ldots, x_n$, let μ_n be the mean of Σ_n, and let σ_n^2 be the variance of Σ_n. Under some general, easily met conditions, as n, the number of random variables, goes to infinity, the distribution function of the random variable

$$\frac{\Sigma_n - \mu_n}{\sigma_n} \qquad \text{Eq. 2.3}$$

approaches the standard normal distribution function.

■

In practice, the number of random variables summed never goes to infinity. The normal approximation is usually considered to be reasonably good for a large n, i.e., $n > 30$ (Conover,

1999). In general, if the variables x_i are identically distributed, and the distribution of each x_i does not depart radically from the normal distribution, then the central limit theorem works quite well even for a small value of random variables (Montgomery, 1991). These conditions are met frequently in monitoring and quality-control problems. Measurements contain random errors due to, e.g., the sampling, the sample pre-treatment, noise, etc. These random errors will tend to a normal distribution as the number of measurements becomes larger.

As a consequence of the central limit theorem we know that regardless of the distribution of the data (population), the sampling distribution of the sample mean is approximately normal. So it is expected that, for a stationary dynamic process under nominal operation, some stochastic signals (features and residuals) used for fault detection and diagnosis purposes follow approximately a normal distribution.

2.3.4 Univariate Statistical Monitoring

A univariate statistical approach to limit sensing can be used to determine the thresholds for each observation variable (a process variable observed through a sensor reading); these thresholds define the boundary for nominal ("in-control") operations, and a violation of these limits with on-line data would indicate a fault. This approach is typically employed using a Shewhart control chart (Fig. 2.1), and has been referred to as limit sensing, and limit value checking (Chiang, et al., 2001).

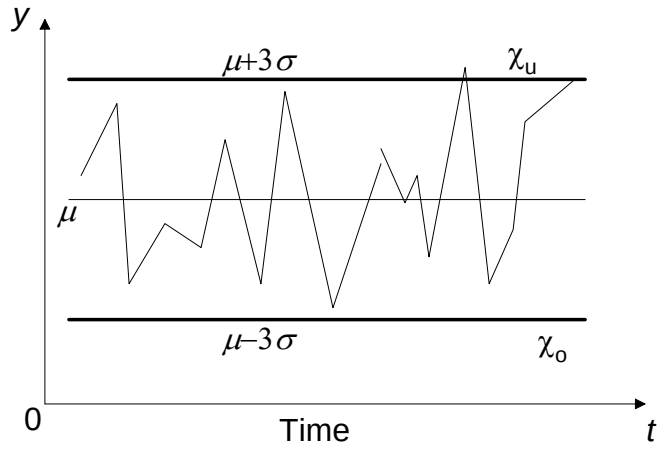


Fig. 2.1 - Typical Shewhart control chart for a dynamic process.

The values of the upper and lower thresholds (χ_u and χ_o) on the Shewhart chart are critical to minimizing the rate of false alarms, and the rate of missed fault detections.

A false alarm is an indication of a fault, when in fact a fault has not occurred. A missed detection is no indication of a fault, though a fault has occurred. For fault detection, there is an inherent trade-off between minimizing the false alarm and missed detection rates.

Given certain threshold values, statistical hypothesis theory can be applied to predict the false alarm and missed detection rates based on the statistics of the data in the training sets (Chiang, et al., 2001). Assuming for a process variable y that any deviations from a desired value μ are due to inherent measurement and process variability described by a normal distribution $N(\mu, \sigma^2)$ with standard deviation σ, then the probabilities that y is in certain intervals are given by:

$$\Pr\{y < (\mu - c_{\alpha/2}\sigma)\} = \Pr\{y > (\mu + c_{\alpha/2}\sigma)\} = \alpha / 2 \qquad \text{Eq. 2.4}$$

$$\Pr\{(\mu - c_{\alpha/2}\sigma) \le y \le (\mu + c_{\alpha/2}\sigma)\} = 1 - \alpha$$

In Eq. 2.4, $c_{\alpha/2}$ is the standard normal deviate corresponding to the 1 - α/2 percentile, α being the level of significance, which specifies the degree of trade-off between the false alarm rate and the missed detection rate. Some typical standard normal deviate values are {3.0; 2.81; 2.58} (Chiang, et al., 2001; Montgomery, 1991). For a standard normal deviate $c_{\alpha/2} = 3$ the probability $\Pr\{(\mu - c_{\alpha/2}\sigma) \le y \le (\mu + c_{\alpha/2}\sigma)\} = 99.73$ %. This value (3-sigma) is used in this work, and is termed 3-sigma (control) limit for thresholds in monitoring tasks, based on a typical Shewhart control chart.

Process monitoring schemes based on Shewhart charts may not provide adequate false alarm and missed detection rates. These rates can be improved by employing measures that incorporate observations from multiple consecutive time instants, such as the cumulative sum (CUSUM) and the exponentially-weighted moving average (EWMA) charts (Montgomery, 1991). For a given false alarm rate, these methods can increase the sensitivity to faults over the measures using the Shewhart charts and decrease the missed detection rate. This is achieved at the expense of increasing the detection delay, which is the amount of time expended between the start of the fault and time of detection. This suggests that the Shewhart charts are better suited for detecting abrupt large process shifts, and the CUSUM and EWMA charts are better for faults producing small persistent process shifts (Chiang, et al., 2001).

The univariate statistical charts (Shewhart, CUSUM, and EWMA) determine the thresholds for each observation variable individually without considering the information contained in the other variables. Because these methods ignore the correlation between variables, they do

not accurately characterize the behaviour of most modern industrial processes (Chiang, et al., 2001). The next section describes the multivariate *T2* statistics, which takes into account the correlations between the observation variables.

2.3.5 Multivariate *T2* Statistics

Let the data in the training set, consisting of *m* observation variables and *n* observations for each variable, be stacked into a data matrix $\mathbf{X} \in \Re^{n \times m}$, then the sample covariance matrix of the training set is equal to

$$\mathbf{S} = \frac{1}{n\text{-}1} \mathbf{X}^{\mathrm{T}} \mathbf{X} . \qquad \text{Eq. 2.5}$$

An eigenvalue decomposition of the matrix $\mathbf{S} \in \Re^{m \times m}$ obtained by singular value decomposition (SVD),

$$\mathbf{S} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{\mathrm{T}} \qquad \text{Eq. 2.6}$$

reveals the correlation structure for the covariance matrix, where $\mathbf{\Lambda} \in \Re^{m \times m}$ is a diagonal eigenvalue matrix and $\mathbf{V} \in \Re^{m \times m}$ is an orthogonal eigenvector matrix, i.e. $\mathbf{V}^{\mathrm{T}} \mathbf{V} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix (Chiang, et al., 2001; Golub & Loan, 1996). The projection $\mathbf{y} = \mathbf{V}^{\mathrm{T}} \mathbf{x}$ of an observation vector $\mathbf{x} \in \Re^{m \times 1}$ decouples the observation space into a set of uncorrelated variables corresponding to the elements of $\mathbf{y}$.
Assuming $\mathbf{S}$ is invertible and with the definition

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2} \mathbf{V}^{\mathrm{T}} \mathbf{x} \qquad \text{Eq. 2.7}$$

the Hotelling's *T2* statistics is given by (Chiang, et al., 2001)

$$\mathrm{T}^2 = \mathbf{z}^{\mathrm{T}} \mathbf{z} . \qquad \text{Eq. 2.8}$$

The matrix $\mathbf{V}$ rotates the major axes for the covariance matrix of $\mathbf{x}$, so that they directly correspond to the elements of $\mathbf{y}$, and $\mathbf{\Lambda}$ scales the elements of $\mathbf{y}$ to produce a set of variables with unit variance corresponding to the elements of $\mathbf{z}$.

The *T2* statistics is a scaled squared 2-norm of an observation vector **x** from its mean. The scaling on **x** is in the direction of the eigenvectors and is inversely proportional to the standard deviation along the eigenvectors. This allows a scalar threshold to characterize the variability of the data in the entire *m*-dimensional observation space. Appropriate thresholds for the *T2* statistics based on the level of significance α can be determined by assuming that the observations are randomly sampled from a multivariate normal distribution. When the actual covariance matrix for the nominal region is estimated from the sample covariance matrix (Eq. 2.5), faults can be detected for observations taken outside the training set using the threshold given by

$$T_{\alpha}^{2} = \frac{m\,(n\text{-}1)\,(n\text{+}1)}{n\,(n\text{-}m)} F_{\alpha}(m,\, n\text{-}m) \qquad \text{Eq. 2.9}$$

where $F_{\alpha}(m, n\text{-}m)$ is the Fisher's *F*-distribution with *m* and *n-m* degrees of freedom (Montgomery, 1991). Given two sample vectors originating from two chi-square distributions, the *F*-distribution is given by the ratio of the estimated variances of the two distributions. The set $T^2 \le T_{\alpha}^2$ is an elliptical confidence region in the observation space. The threshold given by Eq. 2.9 assumes that the observation at one time instant is statistically independent to the observations at other time instants. This can be a misleading assumption for short sampling periods. However, if there is enough data in the training set to capture the normal process variations, the *T2* statistics can be an effective tool for process monitoring, even if there are smooth deviations from the normality or statistical independence assumptions.

The above *T2* tests are multivariable generalizations of the Shewhart chart used in the scalar case. The single variable CUSUM and EWMA charts can also be generalized to the multivariable case in a similar manner.

The quality and quantity of the data in the training set have a large influence on the effectiveness of the *T2* statistics as a process monitoring tool. The required number of observations needed to statistically populate the covariance matrix for *m* observation variables is approximately 10 times the dimensionality of the observation space ($n \cong 10\,m$),(Chiang, et al., 2001). The number of observations *n* is given by the amount of data needed to produce a threshold value sufficiently close to the threshold obtained by assuming infinite data in the training set.

2.3.6 Principal Component Analysis (PCA)

In processes where redundancy or correlation between variables exists, it is advantageous to reduce the number of variables, maintaining an important quantity of original information. Dimensionality reduction techniques can greatly simplify and improve process monitoring procedures, since they project the data into a lower-dimensional space that accurately characterizes the state of the process (Chiang, et al., 2001; Jackson, 2003; Jolliffe, 2002).

Principal Component Analysis (linear PCA) is one of the most popular dimensionality reduction techniques. PCA is a multivariate statistical technique in which a number of related variables are transformed to a smaller set of uncorrelated variables. PCA is also known as Empirical Orthogonal Function Analysis (EOFA). The terminology Karhunen-Loeve Transform (KLT), or expansion, is in common use in some disciplines to denote PCA in continuous time domain.

PCA preserves the correlation structure between the process variables, and captures the variability in the data. The application of linear PCA as a dimensionality reduction tool for monitoring industrial processes has been studied by several academic and industrial engineers. For many applications, most of the variability in the data can be captured in two or three dimensions, and the visualization can be done on a single plot (Chiang, et al., 2001). This one-plot visualization assists the operators and engineers in interpreting the significant trends of the process behaviour.

PCA can produce lower-dimensional representations of data, and therefore, improve the proficiency of detecting and diagnosing faults. The structure abstracted by PCA can be useful in identifying either the variables responsible for the fault and/or the variables most affected by the fault. PCA can separate the observation space into two subspaces: the signal subspace capturing the systematic trends of the process and the noise subspace containing essentially the random noise.

PCA is a linear dimensionality reduction technique that captures the variability of the data. It determines loading vectors (orthogonal vectors) ordered by the amount of variance explained in the loading vector directions. Given a training set of *n* observations and *m* process variables stacked into a data matrix $\mathbf{X} \in \Re^{n \times m}$, the loading vectors are computed by solving the stationary points of the optimization problem (Chiang, et al., 2001):

$$\max_{\mathbf{v} \neq 0} \left(\frac{\mathbf{v}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \mathbf{v}}{\mathbf{v}^{\mathrm{T}} \mathbf{v}} \right) \qquad \text{Eq. 2.10}$$

where $\mathbf{v} \in \Re^{m \times 1}$. The stationary points of Eq. 2.10 can be computed via a Singular Value Decomposition (SVD)

$$\frac{1}{\sqrt{n\text{-}1}} \mathbf{X} = \mathbf{U}\, \mathbf{\Sigma}\, \mathbf{V}^{\mathrm{T}} \qquad \text{Eq. 2.11}$$

where $\mathbf{U} \in \Re^{n \times n}$ and $\mathbf{V} \in \Re^{m \times m}$ are unitary matrices, and the matrix $\mathbf{\Sigma} \in \Re^{n \times m}$ contains the non-negative real singular values of decreasing magnitude along its main diagonal ($\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min(m,n)} \geq 0$), and zero off-diagonal elements. The loading vectors are the orthonormal column vectors in the matrix **V**, and the variance of the training set projected along the i^{th} column of **V** is equal to σ_i^2. Solving Eq. 2.11 is equivalent to solving an eigenvalue decomposition of the sample covariance matrix **S** (Chiang, et al., 2001),

$$\mathbf{S} = \frac{1}{n\text{-}1} \mathbf{X}^{\mathrm{T}}\, \mathbf{X} = \mathbf{V}\, \mathbf{\Lambda}\, \mathbf{V}^{\mathrm{T}} \qquad \text{Eq. 2.12}$$

where the diagonal matrix $\mathbf{\Lambda} = \mathbf{\Sigma}^{\mathrm{T}}\, \mathbf{\Sigma}$ ($\mathbf{\Lambda} \in \Re^{m \times m}$) contains the non-negative real eigenvalues of decreasing magnitude ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$) and the i^{th} eigenvalue equals the square of the i^{th} singular value (*i.e.*, $\lambda_i = \sigma_i^2$).

When the goal is to minimize the effect of random noise that corrupt the PCA representation, and to optimally capture the variations of data, then only the loading vectors corresponding to the *a* largest singular values must be retained in the PCA model. PCA projects the observation space into two subspaces: the scores subspace and the residual subspace. Selecting the columns of the loading matrix $\mathbf{P} \in \Re^{m \times a}$ to correspond to the loading vectors $\mathbf{V} \in \Re^{m \times m}$ associated with the *a* largest singular values, the projections of the observation data $\mathbf{X} \in \Re^{n \times m}$ into the lower-dimensional space are contained in the scores matrix $\mathbf{T} \in \Re^{n \times a}$

$$\mathbf{T} = \mathbf{X}\, \mathbf{P} \qquad \text{Eq. 2.13}$$

and the projection of **T** back into the *m*-dimensional observation space is given by

$$\hat{\mathbf{X}} = \mathbf{T}\, \mathbf{P}^{\mathrm{T}} . \qquad \text{Eq. 2.14}$$

The difference between $\mathbf{X}$ and $\hat{\mathbf{X}}$ is the residual matrix $\mathbf{E}$,

$$\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}} \qquad \text{Eq. 2.15}$$

that captures the variations in the observation space spanned by the loading vectors associated with the *m-a* smallest singular values. The subspaces spanned by $\hat{\mathbf{X}}$ and $\mathbf{E}$ are usually denominated scores space and residual space, respectively. A more accurate representation of the process is given by $\hat{\mathbf{X}}$, since the subspace contained in the matrix $\mathbf{E}$, that has a small signal-to-noise ratio, is removed.

The Square of Prediction Error (SPE), also known as the *Q* statistics, is computed based on the residual space, and is given by $q(k) \in \Re^{1 \times 1}$ for each time instant *k*, assuming a window of length one:

$$q(k) = \mathbf{e}(k)\, \mathbf{e}(k)^{\mathrm{T}} \qquad \text{Eq. 2.16}$$

where the estimation error is $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$, and $\mathbf{x}(k) = \mathbf{X}(k,:) \in \Re^{1 \times m}$ is a line vector.
The distribution of the *Q* statistics has been approximated by Jackson & Mudholkar (1979):

$$Q_\alpha = \theta_1 \left[\frac{h_0\, c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \qquad \text{Eq. 2.17}$$

where $\theta_i = \sum_{j=a+1}^{m} \lambda_j^i$, $h_0 = 1 - \frac{2\,\theta_1\,\theta_3}{3\,\theta_2^2}$, and c_α is the normal deviate corresponding to the $(1-\alpha)$ percentile. Given a level of significance, α, the threshold for the *Q* statistic can be computed using Eq. 2.17 and used to detect faults.

Defining $\mathbf{t}_i$ to be the i^{th} column of $\mathbf{T}$ in the training set, the following properties can be shown (Piovoso & Kosanovich, 1994):

1. $\mathrm{Var}(\mathbf{t}_1) \geq \mathrm{Var}(\mathbf{t}_2) \geq \ldots \geq \mathrm{Var}(\mathbf{t}_a)$, where Var(.) means variance.

2. $\mathrm{Mean}(\mathbf{t}_i) = 0;\ \forall i$

3. $\mathbf{t}_i^{\mathrm{T}} \mathbf{t}_j = 0; \forall\, i \neq j$

4. There is no other orthogonal expansion of a components that captures more variations of the data.

A new observation (column) vector in the testing set for a given time instant k, $\mathbf{x}(k) \in \Re^{m \times 1}$, can be projected into the lower-dimensional scores space $\mathbf{t}_i = \mathbf{x}^{\mathrm{T}} \mathbf{p}_i$. The i^{th} loading vector is $\mathbf{p}_i$, the transformed observations are the scores $\mathbf{t}_i$, and the transformed variables are the principal components. Using the PCA dimensionality reduction technique, only $a \leq m$ variables needed to be monitored, as compared with m variables without the use of PCA.

The text above introduces the main ideas of classic linear PCA. It is also possible to implement linear PCA using neural networks, and nonlinear PCA based on neural networks or on principal curves.

Since the Principal Components Regression (PCR) method is directly related to SVD and PCA, it will be briefly introduced in the next section. This method belongs to the class of parameter estimation/identification methods.

2.3.7 Principal Components Regression (PCR)

Principal Components Regression (PCR) can be understood as an extension of Principal Component Analysis (PCA) to the modelling of some **Y** data from the **X** data (Geladi & Kowalsky, 1986; Wise & Ricker, 1990; Piovoso & Kosanovich, 1996). The approach to defining this relationship is accomplished in two steps. The first is to perform PCA on the **X** data, and the second is to regress the scores onto the **Y** data.

According to Wise & Ricker (1990) parameterized models identified by classical least-squares (LS) are generally as good as models identified by PCR. The exception to this is when conditions are extremely adverse, e.g. there is poor input excitation in the data and the noise level is very high. The concept of input persistent excitation is particularly important in noisy systems. The input signals must verify the Persistent Excitation Conditions (PEC) when it is desired to obtain good consistency of the parameter estimates (Soderstrom & Stoica, 1989; Ljung, 1999).

PCR can be understood as an extension of LS and PCA. In the first step, the principal components are computed. All the scores or the most important for principal components are used as the basis for the LS with the target data **y**. The main advantages of PCR over LS are: a) the noise remains in the residuals, since the eigenvectors with low eigenvalues represent only parts of the data with low variance; b) the regression coefficients **b** are more stable, due

to the fact that the eigenvectors are orthogonal to each other. The great disadvantage of PCR over LS is that it does not have a simple recursive version like the recursive least squares (RLS) algorithm (Soderstrom & Stoica, 1989; Mosca, 1995).

The Parameter Estimation Problem for the case of ARX models can be formulated as follows. Assuming that for a time series of data the vector of the output variable is $\mathbf{y} \in \Re^{n \times 1}$, and the input matrix $\mathbf{X} \in \Re^{n \times m}$ consists of past values of process outputs and inputs, the estimation problem can be expressed in the form (Eq. 2.18), where $\mathbf{b} \in \Re^{m \times 1}$ is the parameter vector.

$$\mathbf{y} = \mathbf{X}\,\mathbf{b}. \qquad \text{Eq. 2.18}$$

The most obvious way to estimate the parameter vector **b** is by least squares (LS) (Soderstrom & Stoica, 1989; Ljung, 1999; Wise, 1991):

$$\hat{\mathbf{b}} = (\mathbf{X}^T\,\mathbf{X})^{-1}\,\mathbf{X}^T\,\mathbf{y}\,. \qquad \text{Eq. 2.19}$$

The main problem with the LS approach is that the matrix $(\mathbf{X}^T\,\mathbf{X})^{-1}$ may be poorly conditioned. If the independent data block is nearly rank deficient (the covariance matrix has some eigenvalues near zero), then the solution to the normal regression equation can change drastically with just a small change in data, for example due to corruption from noise.

To obtain a better conditioned problem for the estimation of the parameter vector **b** it is common to use Principal Components Regression (Wise & Ricker, 1990; Piovoso & Kosanovich, 1996). Either PCA or SVD can be used to decompose the **X** matrix

$$\mathbf{X} = \mathbf{T}\,\mathbf{P}^T = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^T \qquad \text{Eq. 2.20}$$

where $\mathbf{T} = \mathbf{U}\,\mathbf{S}$ and $\mathbf{P}^T = \mathbf{V}^T$. Here, the PCA decomposition will be used, where the scores matrix **T** has orthogonal columns and the loadings matrix $\mathbf{P}^T$ has orthonormal columns. The vectors in $\mathbf{P}^T$ are arranged in such a way that each lies in the direction of greatest remaining variance in **X**, after the variance in the previous vector directions is removed. The PCA decomposition provides a new basis set $\mathbf{P}^T$ for the **X** matrix, where each successive basis vector describes the remaining trend in the **X** matrix.

For the case of SISO systems and without loss of generality, the estimation problem via PCR is formulated here as a theorem inspired in previous algorithm formulations (Geladi & Kowalsky, 1986; Wise & Ricker, 1990; Piovoso & Kosanovich, 1996). A proof is also given.

Theorem 2. Principal Components Regression (PCR) theorem.

For the parameter estimation problem $\mathbf{y} = \mathbf{X}\,\mathbf{b}$ (Eq. 2.18), let a data matrix $\mathbf{X} \in \Re^{n \times m}$ consists of past values of process outputs $\mathbf{y} \in \Re^{n \times 1}$ and inputs $\mathbf{u} \in \Re^{n \times 1}$. The number of parameters to estimate is given by m, and the number of principal components is given by $a \leq m$. If n >> m and the persistent excitation conditions are verified then it is feasible to decompose the $\mathbf{X}$ matrix by Principal Component Analysis (PCA). Estimates of the parameter vector $\mathbf{b}$ ($\mathbf{b} = \boldsymbol{\theta}$) can be obtained using the first a vectors in $\mathbf{P}^T$ (principal components, or PCs) and in $\mathbf{T}$ according to

$$\hat{\mathbf{b}} = \mathbf{P}_a\,(\mathbf{T}_a{}^T\,\mathbf{T}_a)^{-1}\,\mathbf{T}_a{}^T\,\mathbf{y} \qquad \text{Eq. 2.21}$$

with $\hat{\mathbf{b}} \in \Re^{m \times 1}$, $\mathbf{P}_a \in \Re^{m \times a}$, and $\mathbf{T}_a \in \Re^{n \times a}$.

■

Proof. If the PCA decomposition of matrix $\mathbf{X}$, $\mathbf{X} = \mathbf{T}_a\,\mathbf{P}_a{}^T$ (Eq. 2.20), is substituted in the least-squares equation $\hat{\mathbf{b}} = (\mathbf{X}^T\,\mathbf{X})^{-1}\,\mathbf{X}^T\,\mathbf{y}$ (Eq. 2.19), then Eq. 2.21 is obtained, provided that $(\mathbf{P}_a\,\mathbf{P}_a{}^T)^{-1} = \mathbf{I}$, according to the following developments:

$$\hat{\mathbf{b}} = (\mathbf{X}^T\,\mathbf{X})^{-1}\,\mathbf{X}^T\,\mathbf{y} \quad \text{and} \quad \mathbf{X} = \mathbf{T}_a\,\mathbf{P}_a{}^T\,;$$

$$\hat{\mathbf{b}} = ((\mathbf{T}_a\,\mathbf{P}_a{}^T)^T\,(\mathbf{T}_a\,\mathbf{P}_a{}^T))^{-1}\,(\mathbf{T}_a\,\mathbf{P}_a{}^T)^T\,\mathbf{y} =$$

$$= ((\mathbf{P}_a\,\mathbf{T}_a{}^T)(\mathbf{T}_a\,\mathbf{P}_a{}^T))^{-1}\,(\mathbf{P}_a\,\mathbf{T}_a{}^T)\,\mathbf{y} =$$

$$= (\mathbf{P}_a\,\mathbf{P}_a{}^T)^{-1}\,\mathbf{P}_a\,(\mathbf{T}_a{}^T\,\mathbf{T}_a)^{-1}\,\mathbf{T}_a{}^T\,\mathbf{y} =$$

$$= \mathbf{P}_a\,(\mathbf{T}_a{}^T\,\mathbf{T}_a)^{-1}\,\mathbf{T}_a{}^T\,\mathbf{y}$$

■

Note that the PCR problem is well conditioned. Since the score vectors in $\mathbf{T}$ are orthogonal, the matrix $(\mathbf{T}_a{}^T\,\mathbf{T}_a)$ is non-zero on the diagonal only, and its inverse is easily calculated. The main idea in the PCR estimation is to invert only the major trends in $\mathbf{X}$ when estimating $\mathbf{b}$, assuming implicitly that the major trends in $\mathbf{X}$ and in $\mathbf{y}$ are causally linked. In practice, when using the PCR estimation method it is common to use $a = m$ when good estimates are desirable, since generally as principal components are added the model error gets smaller, going through a minimum (Geladi & Kowalsky, 1986; Wise & Ricker, 1990).

A sliding window Principal Components Regression (PCR) algorithm is proposed here to compute on-line an estimation of the ARX parameter vector. The sliding window length d must be selected according to the dominant process time constant. The pseudo-code of the algorithm is given afterwards. A comparison of the algorithms RLS, sliding window LS and sliding window PCR is presented in section 5.5.

In a fault situation, one sliding window algorithm with finite time horizon, like the SW-PCR, guarantees that a time instant exist from which all the information reflects the fault behaviour.

Algorithm 1. Sliding Window Principal Components Regression (SW-PCR).

Notation:

k: current time instant;

T_s: sampling period;

d: window length in seconds; w: window length in samples;

n_a, n_b, n_d: ARX model orders;

$\boldsymbol{\theta}(k)$: ARX parameter vector;

$\mathbf{u}$: $[u(1) \ldots u(k)]$: vector of input signal;

$\mathbf{y}$: $[y(1) \ldots y(k)]$: vector of output signal;

a: number of principal components;

Initialization:

$w = \text{round}(d / T_s)$; $a = n_a + n_b$; $n_c = n_d + n_b - 1$;

0. at time instant k:

1. $k_a = k - w + 1$; // first sample of sliding window

2. $\mathbf{y}_v = \mathbf{y}(k_a{:}k)$; // vector of output signal for a sliding window

3. $\mathbf{X} = [-\mathbf{y}(k_a\text{-}1{:}k\text{-}1) \ldots -\mathbf{y}(k_a\text{-}n_a{:}k\text{-}n_a) +\mathbf{u}(k_a\text{-}n_d{:}k\text{-}n_d) \ldots +\mathbf{u}(k_a\text{-}n_c{:}k\text{-}n_c)]$; // data matrix

4. $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{X})$; // singular value decomposition of $\mathbf{X}$

5. $\mathbf{T} = \mathbf{U}\,\mathbf{S}$; // scores matrix

6. $\mathbf{P} = \mathbf{V}$; // loadings matrix

7. $\mathbf{P}_a = \mathbf{P}(:,1{:}a)$; // loadings matrix associated with a principal components

8. $\mathbf{T}_a = \mathbf{T}(:,1{:}a)$; // scores matrix associated with a principal components

9. $\boldsymbol{\theta}(k) = \mathbf{P}_a\,(\mathbf{T}_a^{\mathrm{T}}\,\mathbf{T}_a)^{-1}\,\mathbf{T}_a^{\mathrm{T}}\,\mathbf{y}_v$; // vector of parameters estimated at time k

10. $\mathbf{r}_g(k) = [-y(k\text{-}1) \ldots -y(k\text{-}n_a) +u(k\text{-}n_d) \ldots +u(k\text{-}n_c)]$; // regressor vector at time k

11. $y_p(k) = \mathbf{r}_g(k)^{\mathrm{T}}\,\boldsymbol{\theta}(k)$; // output predictor value

■

It is worth understanding the relationship between SVD, PCA, and eigenvector decompositions. The principal components $\mathbf{P}^T$ are equal to the singular vectors $\mathbf{V}^T$, and these are also equal to the eigenvectors of ($\mathbf{X}^T\ \mathbf{X}$). Besides, the singular values of $\mathbf{X}$ are equal to the square roots of the eigenvalues of ($\mathbf{X}^T\ \mathbf{X}$).

A relation also exists between the covariance matrix and the correlation matrix (Chiang, et al., 2001; Wise, 1991). If the variables (columns) in the data matrix $\mathbf{X}$ have been mean centered (scaled to have zero mean), then ($\mathbf{X}^T\ \mathbf{X}$) / (n - 1) (where "n" equals the number of samples) is referred to as the covariance matrix of $\mathbf{X}$. If the data matrix $\mathbf{X}$ is auto-scaled (a standardization transformation is made in order for columns scaled to have zero mean and unit variance) then ($\mathbf{X}^T\ \mathbf{X}$) / (n - 1) is mentioned as the correlation matrix. The eigenvectors of the correlation matrix are equal to the principal component vectors.

2.3.8 Pattern Recognition and Discriminant Analysis

Today's industrial processes are heavily instrumented, with a large amount of data collected on-line and stored in computer databases. If the data collected during nominal operation and faulty situations have been previously diagnosed, then the data can be categorized into separate classes where each class is associated to a particular fault (Chiang, et al., 2001).

If hyper-surfaces can separate the data in different classes, these hyper-surfaces can define the decision boundaries for each of the fault regions. In general the decision boundaries are nonlinear, and this is the reason why here they are named hyper-surfaces. Once a fault is detected using on-line data observations, the fault can be diagnosed by determining the fault region in which the observations are located, assuming the fault is represented in the database. The assignment of data to one of several classes (or categories) is the problem addressed by Pattern Recognition theory (Marques de Sá, 2001; Duda, et al., 2000; Salvador Marques, 1999). The human perception is a paradigm in the area of Pattern Recognition. The pattern recognition systems try to emulate human capabilities, and their main goal is to take a decision between a set of classes.

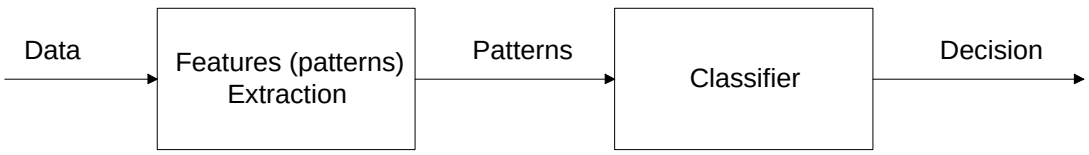


Fig. 2.2 - Classical structure of a pattern recognition system.

The classical structure of a pattern recognition system is composed of two blocks (Salvador Marques, 1999), as depicted in Fig. 2.2. The first block is a features (patterns) extraction block, and the second is a classifier. The first block, selects the important information for the decision, transforming the data into a set with fewer values named features or patterns. Finally, the patterns are used by the classifier to select the class, or hypothesis, that best describes the real world.

Typically, it is assumed that the extracted patterns (features) belong to a finite set of discrete features, for example, $S_d = \{1, 2, \ldots, M\}$, or a set $S_c \in \Re^n$ of continuous features. For the first case, the observed pattern is a scalar contained in the set S_d, while for the second case the pattern is a vector with n real components, $\mathbf{v} = [v_1\ v_2\ \ldots\ v_n]^T$. The set S is often termed the space of patterns (features). The main goal of the classifier block is to assign a class $c \in \Omega$ to each observed pattern, where $\Omega = \{c_1, c_2, \ldots, c_m\}$ is the set of admissible classes, and $m = \#\Omega$.
The typical pattern recognition system assigns an observation vector to one of several classes via three steps: feature extraction, discriminant analysis, and maximum selection. The classifier executes the steps of discriminant analysis and maximum selection. Several approaches exist to solve the problem of pattern recognition. The most common are classical statistical parametric approaches, and neural approaches.
For the linear case, the Discriminant Analysis can be implemented based on Fisher Discriminant Analysis (FDA) (Chiang, et al., 2001). FDA is a linear statistical dimensionality reduction technique that has been extensively studied in the pattern classification literature. It takes into account the information between the classes and has advantages over PCA for fault diagnosis. FDA is optimal in terms of maximizing the separation amongst the classes. For the nonlinear case, the Nonlinear Discriminant Analysis (NLDA) can be implemented using neural networks (Asoh & Otsu, 1990; Gallinari, et al., 1991). In the neural network training phase, the minimization of the performance functions (sum of squared errors SSE, or mean of squared errors MSE) corresponds to a maximization of the discriminant criterion (Gallinari, et al., 1991). The neural NLDA (NNLDA) is used in this work, and will be explained in Chapter 3 (section 3.3) in a fault detection and diagnosis context. The discriminant analysis NNLDA allows the definition of decision boundaries needed for fault detection and isolation, and is more efficient than the geometrical techniques.

2.4 Dynamic Systems and Black-Box Models

2.4.1 Introduction

The concepts of Signals and Systems arise in an extremely wide range of fields, and the ideas and techniques associated with these concepts play an important role in such diverse areas of science and technology (Oppenheim, et al., 1983). Here is assumed that systems are dynamic, since the work developed is focused on technical processes. The main focus is on sampled data systems (data captured from continuous time systems), and the concepts proposed were developed in discrete time.

Inferring Models from observations and studying their properties is really what science is about. The models (”laws of nature”, “hypothesis”, “paradigms”, etc) may be of more or less formal character, but they have the basic feature that they attempt to link observations together into some pattern (Ljung, 1999).

2.4.2 Dynamic Systems, Signals and Models

Sometimes, it is a good idea to remember and to write elementary definitions, since more complex concepts are based on them, and they can also be beneficial to develop new concepts. Clearly the notions of a system and of a signal are broad concepts, and it is not surprising that they play an important role in modern science (Ljung, 1999).

Definition 2.3. Signal.

A signal is a function of one or more independent variables, and typically contains information about the behaviour or nature of some phenomenon (Oppenheim, et al., 1983).

■

Definition 2.4. System.

A system responds to particular input signals by producing other output signals (Oppenheim, et al., 1983). Another definition is: a group of interacting, interrelated, or interdependent elements forming a complex whole. In loose terms, a system is an object in which variables of different physical nature interact and produce observable signals (Ljung, 1999).

■

Usually, the interesting observable signals are the output signals **y**(*k*). The system is also affected by external signals. The external signals that can be manipulated by the operator are called input signals **u**(*k*). Others are called disturbances, and can be divided into those that are directly measured **w**(*k*) and those that are only observed through their influence on the system output **v**(*k*) (Fig. 2.3).

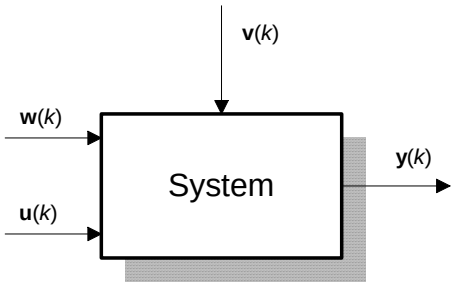


Fig. 2.3 - A system with inputs, disturbances and outputs.

When we interact with a system, we need some concept of how its signals (variables) relate to each other.

Definition 2.5. System model.
A model of a system is an assumed relationship among observed signals (Ljung, 1999).
■

System models may come in various shapes and be stated with varying degrees of mathematical formalism. The intended use will determine the degree of sophistication that is required to make the model purposeful. A model for fault detection and diagnosis does not need to be equal to a model for control; it can be simpler, or more complex.

Basically, there are two ways of constructing mathematical models (Ljung, 1999; Soderstrom & Stoica, 1989): physical (mathematical) modeling, and system identification. Physical modeling is an analytical approach based on basic laws from physics of the phenomenon or process. System identification is an experimental approach. Based on experiments, a model (usually a black-box model) is fitted to the captured data by assigning suitable numerical values to its parameters. System identification is relevant in many applications, e.g., simulation, prediction, control systems design, and fault detection and diagnosis.

The real systems are objects of a different kind than physical models. Taking a pragmatic view of models, the modeling should be guided by “usefulness” rather than “truth” (Ljung, 1999).

The terminology used by the control scientists and engineers is not consensual. The terms Estimator, Predictor, and Observer, are in some situations used improperly. Some definitions used in the work in the field of dynamic systems are presented below. An output signal of a dynamic process is a signal directly measured by a sensor.

Definition 2.6. Estimator.
An estimator is a system that tries to estimate (approximately) the value of a parameter (or a set of parameters) of a model associated with a stochastic process. The parameters being estimated can assume a constant value, or not. If the parameters change, then the estimator must include a re-tuning mechanism; in this case, the estimator tries to estimate the instant value of the parameter. The estimator does not know, a priori, any intrinsic dynamic characteristic of the variation of the parameters.
■

Definition 2.7. Predictor.
A predictor is a system that reproduces the evolution of a measurable signal, with temporal anticipation (or in synchronism) relative to the real signal. The predictor structure can incorporate, or not, a feedback mechanism. A predictor incorporates a dynamic model of the process in question.
■

Definition 2.8. Observer.
An observer is a system that reproduces the evolution of a non-mensurable signal, with temporal anticipation (or in synchronism) relative to the real signal. The observer structure can incorporate, or not, a feedback mechanism. An observer incorporates a dynamic model of the process in question.
■

Both predictors and observers try to reproduce the evolution of signals associated with dynamic processes, and their structure incorporates a dynamic model that takes into account the dynamic characteristics of the process.

Other definitions can be found in (Ljung, 1999; Soderstrom & Stoica, 1989).
In some literature the term observer is used as a structure that incorporates the predictor structure.

2.4.3 Black-Box Models and System Identification

In this work, the proposed fault detection and diagnosis (FDD) and fault tolerance methodologies are based mainly on black-box system models.
Linear ARX models and nonlinear ARX neural models are used in this work. These models are especially important due to their high potential and simplicity.
Some definitions are presented next.

Definition 2.9. Black-Box Model.
A model, whose parameters are basically viewed as vehicles for adjusting the fit to the data and do not reflect physical considerations in the system, is called a black-box model (Ljung, 1999).
■

Input-output models were considered, instead of state-space models, since the proposed FDD and FTC methods are based on this type of models. It is assumed that the systems under study can be modeled by stochastic processes. The concept of a stochastic process is presented below. Important contributions in this area were made by the mathematician Kolmogorov around 1930.

Definition 2.10. Stochastic Process.
A stochastic process (random process) can be regarded as a family of stochastic variables $x(k)$, $k \in T$ (Astrom & Wittenmark, 1997). The stochastic variables are indexed with the parameter k, which belongs to the set T, called the index set. In stochastic control theory, the variable k is interpreted as discrete time. For sampled-data systems, the set T contains the sampling instants, that is, $T = \{1, 2,\ldots, k\text{-}1, k, k\text{+}1, \ldots\}$.
■

Here, two types of black-box models are mainly used and discussed: the autoregressive linear ARX model, and the nonlinear ARX neural model.

Definition 2.11. ARX Model.

An ARX linear model is an input-output parametric model that can be expressed by a linear difference equation (Ljung, 1999):

$$y(k) + a_1\, y(k\text{-}1) + \ldots + a_{na}\, y(k\text{-}n_a) = b_1\, u(k\text{-}n_d) + \ldots + b_{nb}\, u(k\text{-}(n_d\text{+}n_b\text{-}1)) + e(k)\ . \quad \text{Eq. 2.22}$$

Introducing the polynomials $A(q^{-1})$ and $B(q^{-1})$ given by $A(q^{-1}) = 1 + a_1\, q^{-1} + \ldots + a_{na}\, q^{-na}$ and $B(q^{-1}) = b_1\, q^{-nd} + b_2\, q^{-(nd+1)} + \ldots + b_{nb}\, q^{-(nd+nb-1)}$, the ARX model can be written in the form $A(q^{-1})\, y(k) = B(q^{-1})\, u(k) + e(k)$, and its predictor can be expressed in the form $\hat{y}(k|\theta) = B(q^{-1})\, u(k) + [1 - A(q^{-1})]\, y(k)$, or

$$\hat{y}(k|\theta) = \varphi^{T}(k)\, \theta \quad . \qquad \text{Eq. 2.23}$$

The vector of adjustable parameters is expressed by $\theta = [a_1\, a_2 \ldots\ a_{na}\ b_1 \ldots\ b_{nb}]^{T}$, and $\varphi(k) = [-y(k\text{-}1) \ldots -y(k\text{-}n_a)\ +u(k\text{-}n_d)\ + \ldots +u(k\text{-}(n_d\text{+}n_b\text{-}1))]^{T}$ is the regression (data) vector.

The ARX model can be viewed as a possible realization of a stochastic process (Astrom and Wittenmark, 1997). The predictor (Eq. 2.23) defines a linear regression, and this property makes the ARX model a prime choice in many applications.

■

Here, the construction of the black-box ARX model is carried out via System Identification**.** Based on experiments, the black-box model is fitted to the captured data by assigning suitable numerical values to its parameter vector θ. The estimation problem can be viewed as an optimization problem (Ljung, 1999; Soderstrom & Stoica, 1989). The most common parameter estimation methods applied to the determination of θ in linear ARX models are the Least-Squares (LS) method, the Kalman Filter (KF), and the Instrumental Variable (IV) method (Soderstrom & Stoica, 1989; Ljung, 1999). Most of these methods are based on the Kalman filter approach (Haykin, 2002), and recursive versions exist (recursive LS, etc). Another method is the Principal Component Regression (PCR) based on PCA and LS, described in section 2.3.7. These methods require a finite number of parameters, and that is the reason they are called parametric estimation methods. System identification may also be

achieved based on nonparametric estimation methods (Soderstrom & Stoica, 1989): transient analysis, frequency analysis, correlation analysis, spectral analysis, etc.

The ARX model is a suitable model for modeling linear systems. In order to develop Nonlinear Models to model nonlinear dynamic systems, two main approaches can be adopted: 1) a natural approach based on a nonlinear model (white-box model, neural model, fuzzy model, etc); 2) a multi-model approach based on a set of linear models (ARX, state-space, etc). From a theoretical point of view a nonlinear white-box model is more desirable, but in most cases is very hard or even impossible to obtain. When the systems are complex, or hard to model, modelling based on nonlinear black-box models (neural models, fuzzy models, etc) is usually a good and the only alternative.

The artificial Neural Networks (NN) play an important role in many scientific fields such as modelling and identification of dynamic systems, control, pattern recognition, fault detection and diagnosis (Narendra & Parthasarathy, 1990; Norgaard, et al., 2003; Haykin, 1994; Hagan, et al., 1995). A brief history of the development of neural networks, and a broad range of applications areas (automotive, banking and financial, electronics, manufacturing, fault detection and diagnosis, medical, telecommunications, etc), can be found in Hagan, et al., 1995. There does not seem to be one consensual definition of a neural network. The definition of a neural network that can be viewed as an adaptive machine is presented here.

Definition 2.12. Neural Network.

An (artificial) neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects (Haykin, 1994): a) knowledge is acquired by the network through a learning process; b) inter-neuron connection strengths known as synaptic weights are used to store the knowledge.

■

For multilayer perceptron neural networks, the Neuron (Node, or Unit) is a processing element that takes a number of inputs, weights them, sums them up, and uses the result as the argument for the activation function. An illustration of the Neuron Model is depicted in Fig. 2.4 (Norgaard, et al., 2003; Hagan, et al., 1995).

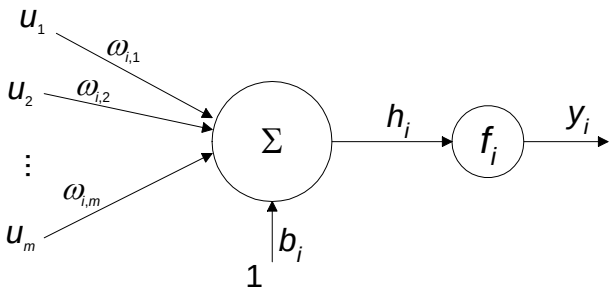


Fig. 2.4 - A neuron model.

Mathematically, the neuron model can be expressed by Eq. 2.24. The inputs to a neuron can either be outputs of other units or they can be external inputs. The displacement b_i is called the bias. The activation function f_i can take any form, but most often it is monotonic. Some common activation functions are: a) the bipolar; b) the linear; c) the Gaussian; d) the hyperbolic tangent sigmoid.

$$y_i = f_i(h_i) = f_i\left(\sum_{j=1}^{m} \omega_{i,j}\, u_j + b_i\right) \qquad \text{Eq. 2.24}$$

When the goal is to obtain models of dynamic nonlinear systems, or applications of process control, the linear activation function $f(x) = x$ and the hyperbolic tangent sigmoid activation function $f(x) = tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$ are commonly adopted. The hyperbolic tangent sigmoid is monotonic, smooth and its range is $-1 \leq f(x) \leq +1$.

Neurons can be combined into a network in various network architectures (Norgaard, et al., 2003; Hagan, et al., 1995; Haykin, 1994). The most common neural network architecture is the feedforward (FF) multilayer perceptron (MLP) neural network (NN).

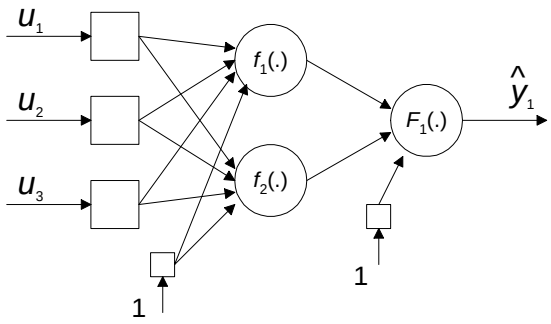


Fig. 2.5 - Feedforward multilayer perceptron neural network (FF-MLP-NN).

Definition 2.13. A feedforward (FF) multilayer perceptron (MLP) neural network (NN).
The basic FF-MLP-NN is constructed by ordering the neurons in layers, letting each neuron in a layer take as inputs only the outputs of neurons in the previous layer or external inputs (Hagan, et al., 1995; Haykin, 1994). Each neuron obeys the model (Eq. 2.24), and its architecture is depicted in Fig. 2.4. A general typical architecture of a FF-MLP-NN, with 3 input signals and 1 output signal, is given in Fig. 2.5.

■

Fig. 2.5 shows a MLP-FF-NN with three layers. The input layer (IL) is a pseudo-layer which receives the input signals. The second layer is the hidden layer (HL) with activation functions $f_i(.)$ in each neuron. Finally, the output layer (OL) has an activation function $F_i(.)$ in each neuron. All the neurons have a bias input.

The training or learning of the network is the task of determining the weights from the examples. It is basically a conventional estimation problem from system identification. That is, the weights are estimated from the examples in such a way that the network, according to some metric, models the true relationship as accurately as possible. In most applications, the training phase is usually carried out in off-line operation, since it commonly requires high computational load and a long time. Some applications need an on-line training. In this work the training is carried out off-line using the Levenberg-Marquardt optimization algorithm (Hagan, et al., 1995). A set of data is used for training and a different set is used for validation. During the training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The most common performance functions used to validate the neural network model are the sum of squared errors (SSE) and the mean of squared errors (MSE). Assuming the vector $\mathbf{p} \in \Re^{1 \times m}$ is the input vector, $\mathbf{t}$ is the target vector, $\mathbf{y}$ is the vector estimated by the neural network, and $\mathbf{e} = \mathbf{t} - \mathbf{y}$ is the prediction error, the performance functions are given by

$$\Gamma_{\text{SSE}} = \mathbf{e}\,\mathbf{e}^{\text{T}} \text{ and } \Gamma_{\text{MSE}} = \frac{1}{m}\sum_{i=1}^{m} e_i^2 \,. \qquad \text{Eq. 2.25}$$

A FF-MLP neural network may be viewed as a practical vehicle for performing a nonlinear input-output mapping of a general nature from a p-dimensional Euclidean input space to a q-dimensional Euclidean output space, $f : \Re^p \Rightarrow \Re^q$ (Haykin, 1994). Many researchers (Cybenko, Funahashi, and Hornik) contribute to demonstrate the Universal Approximation

Theorem that is directly applicable to FF-MLP neural networks. The theorem states that "a single hidden layer is sufficient for a FF-MLP-NN to compute a uniform ε approximation to a given training set represented by the set of inputs $\{u_1,\ldots, u_p\}$ and a desired (target) output $f(u_1,\ldots, u_p)$" (Haykin, 1994). In some sense, the task of capturing a certain input-output relationship encapsulated in the training data by means of a neural network, and via a learning mechanism, can be understood as a nonlinear regression method.

The feed-forward MLP neural networks establish a static transformation (relationship) between the input space $\mathbf{u} \in \Re^p$ and the output space $\mathbf{y} \in \Re^q$. It is possible to introduce dynamics in the FF-MLP-NN if the input vector is composed of past input and output data (Narendra & Parthasarathy, 1990). Next, a definition of a nonlinear ARX (NARX) neural model is presented, assuming that a NARX model is given by

$$y(k) = f(y(k\text{-}1), \ldots, y(k\text{-}n_a), u(k\text{-}n_d), \ldots, u(k\text{-}(n_d+n_b\text{-}1))) , \qquad \text{Eq. 2.26}$$

where *f(.)* is a nonlinear function.

Definition 2.14. NARX Neural Model.

A feed-forward MLP neural network, with sigmoidal activation functions in the hidden layer, is able to approximate any nonlinear function according to the universal approximation theorem described above. Theoretically a FF-MLP-NN is able to model the dynamic behaviour of a nonlinear ARX (NARX) model (Eq. 2.26).

A NARX neural model, written as an output predictor, can be parameterized by

$$\hat{y}(k) = g(\boldsymbol{\varphi}(k),\mathbf{W}) \qquad \text{Eq. 2.27}$$

where $g(.)$ represents a nonlinear transformation due to the neural model, $\mathbf{W}$ congregates the weight matrices, and $\boldsymbol{\varphi}(k) = [y(k\text{-}1), \ldots, y(k\text{-}n_a), u(k\text{-}n_d), \ldots, u(k\text{-}(n_d+n_b\text{-}1))]^T$ is the regression vector at time instant k. According to this approach, the general expression (Eq. 2.27) can be written in the form (Eq. 2.28), where $g(.) \equiv \hat{f}(.)$ represents an estimation of $f(.)$, $y(k)$ and $u(k)$ are the output signal and input signal, respectively, at time instant k.

This NARX neural structure is commonly known as a series-parallel model (Narendra & Parthasarathy, 1990).

$$\hat{y}(k) = g(y(k\text{-}1), \ldots, y(k\text{-}n_a), u(k\text{-}n_d), \ldots, u(k\text{-}(n_d+n_b\text{-}1)), \mathbf{W}) \qquad \text{Eq. 2.28}$$

■

2.4.4 Low Pass Filtering

A first order low pass digital filter $H_{lp}(z, \lambda)$, in discrete time, used in this work is described next. In the area of fault detection and diagnosis, the low pass filtering of fault alarm signals is very important, in order to reduce the false alarm rates. In this work, the fault detection signal is obtained from low pass filtering the fault alarm signal. The fault isolation signal, obtained by statistical or geometrical methods, is usually also low pass filtered.

For a given input signal $v_i(k)$ the infinite impulse response (IIR) low-pass filter $H_{lp}(z)$ computes the output signal $v_f(k)$ according to the difference equation given by

$$v_f(k) = \lambda\, v_f(k\text{-}1) + (1\text{-}\lambda)\, v_i(k) \qquad \text{Eq. 2.29}$$

where λ ($\lambda \geq 0$) is a design parameter (the pole location at the z-plane). The transfer function of the low pass IIR filter, with unitary static gain, is expressed by

$$H_{lp}(z) = \frac{V_f(z)}{V_i(z)} = \frac{1\text{-}\lambda}{1\text{-}\lambda\, z^{-1}}. \qquad \text{Eq. 2.30}$$

In this work, this low pass filter is termed $H_{lp}(z, \lambda)$.

For application in fault detection and isolation the choice of the pole location of the filter (parameter λ) its not a simple task. The pole location λ must be chosen in order to obtain a desired commitment between the rate of false alarms Ψ_{fa}, the rate of missed fault detections Ψ_{fm}, the detection delay d_d or the isolation delay d_i. Since the tasks of fault detection and isolation are separate, the delay is here represented by d, for both tasks. So, the pole location λ can be expressed by the function $\lambda = f_a(\Psi_{fa}, \Psi_{fm}, d)$.

For the case of FDD methods that are based on system identification, like most of the FDD approaches proposed in this work, the delay d is a function of the window length w of the

sliding window parameter estimation algorithm, i.e., $d = f_b(w)$. For this case, the pole location λ can be expressed by the function $\lambda = f_p(\Psi_{fa}, \Psi_{fm}, w)$.

2.5 Supervision, Fault Detection and Diagnosis

2.5.1 Introduction

Research areas related to control engineering and signal processing are in continuous evolution. Due to research, new ideas and concepts emerge, and when reach maturity and a large spectrum allows the creation of new scientific research areas. The Supervision (SPV), Monitoring (MNT), Fault Detection and Diagnosis (FDD), and the Fault Tolerant Control (FTC), are emerging research areas having a strong link with the domain of process control (Isermann, 1997).

For certain critical and industrial systems, the FDD techniques are extremely important since some faults can cause serious failures. Some of the most critical systems are located in nuclear plants, aeronautics, chemical plants, power plants, transportation systems, supplying systems, and communications systems. Nowadays, almost all complex systems incorporate basic fault detection modules. Advanced methods of supervision and fault detection and diagnosis are needed (Isermann, 2004; Frank, et al., 2000a; Chen & Patton, 1999).

Fault tolerance in automatic control systems is gaining more and more importance, and can be achieved either by passive or by active strategies (Blanke, et al., 2003; Patton, 1997).
It is useful to classify the plant faults into three categories: actuator faults, component faults (faults in the framework of the process), and sensor faults, as shown in Fig. 2.6 (Frank, 1996). The faults can commonly be described as input signals. There is always modeling uncertainty due to un-modeled disturbances, noise and model mismatch. This may be not critical for system operation, but may obscure the fault detection by raising false alarms. The modeling uncertainty is usually taken into consideration by vectors of unknown inputs. Faults can also occur in other systems like controllers and supervision systems.

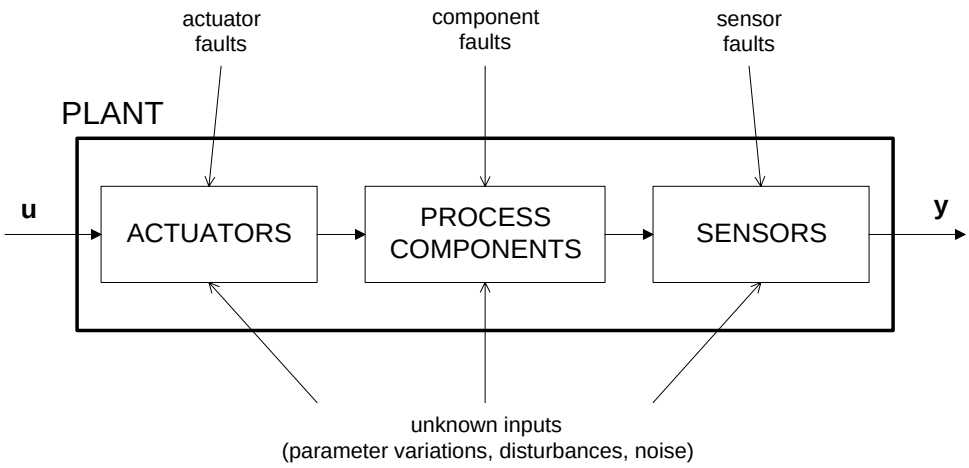


Fig. 2.6 - Definition of faults in a plant.

Fig. 2.7 shows the historic development of the fault detection and diagnosis theory (Frank, et al., 2000b); the work done here is based on the research areas marked. Starting as a special application of observer theory around the year 1970, model based FDD theory went through a dynamic and fast development. Nowadays FDD theory is becoming an important field of automatic control theory. In the first twenty years, it was the control community that made the decisive contributions, while in the last years the trends have been marked by an increasing number of contributions from the computer science and artificial intelligence communities associated with the combination of different FDD approaches.

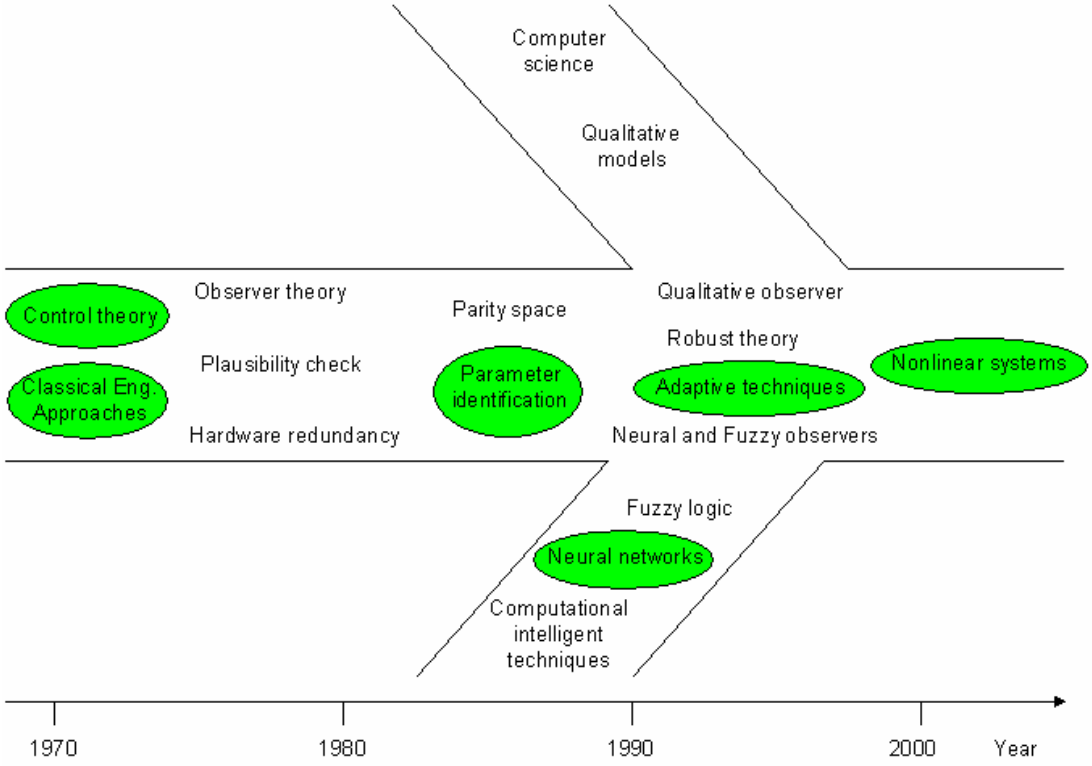


Fig. 2.7 - Historic development of FDD theory.

2.5.2 FDD Terminology and Tasks

The fault detection and diagnosis (FDD) terminology according to the Safeprocess Technical Committee (SPTC) can be found in (Isermann & Balle, 1997). The main definitions of fault, failure, fault detection, fault diagnosis, etc, are resumed in Appendix A. The overall concept of FDD consists mainly of the following two tasks (Isermann, 1997): fault detection and fault diagnosis. The fault diagnosis includes the tasks of fault isolation and fault identification (analysis). In the majority of applications, the FDD approaches only implements the fault detection and isolation (FDI) methods.

A typical model based fault detection and diagnosis (FDD) architecture is depicted in Fig. 2.8 (Isermann, 2004; Moseler & Muller, 2000). Based on measured input and output signals, fault detection approaches generate features using a plant model. The features may include physical or model parameters, residuals, etc. The features computed on-line are compared to the features in the nominal (fault-free) case. The respective deviations are compared to thresholds, and symptoms are obtained. Ideally, for each interesting fault a unique pattern should be associated. The symptoms are processed by a knowledge based system which maps the symptoms to the respective faults. The knowledge can be stored in trained references patterns (classification methods, etc) or in the form of rules (fuzzy rules, etc).

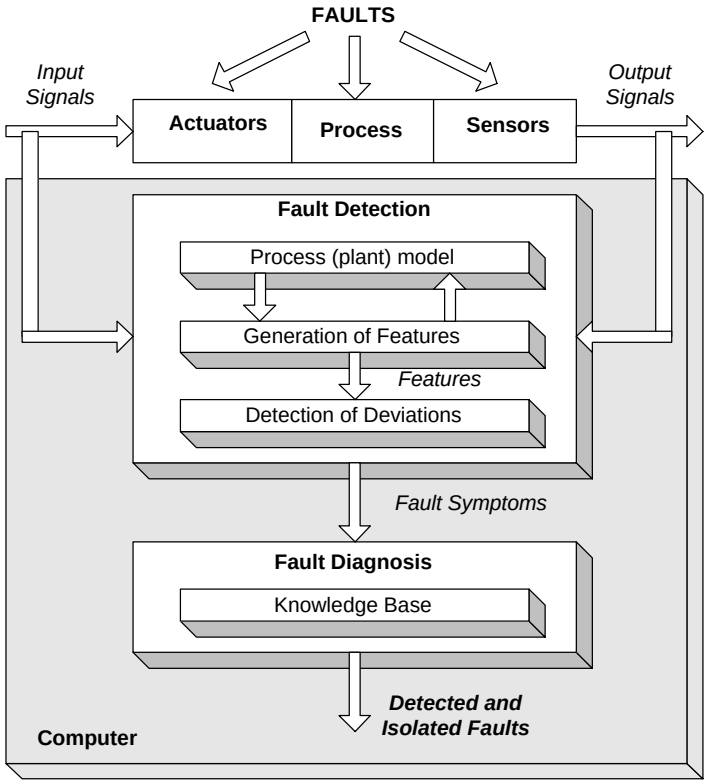


Fig. 2.8 - Typical model-based FDD architecture.

To perform the FDD procedure three steps have to be taken (Frank, 1996). The first step is the residual (symptom) generation (fault detection, FDE), i.e. the generation of signals or symptoms which reflect the faults. The second is the residual evaluation (fault isolation, FIO, or classification), i.e. logical decision-making on the time of occurrence and the location of a fault. The third one is the fault analysis step (fault identification, FID), i.e. determination of the type of fault, size and cause. This three-stage process is illustrated in Fig. 2.9 (Frank, 1996). Note that the first two steps constitute the concept of fault detection and isolation (FDI).

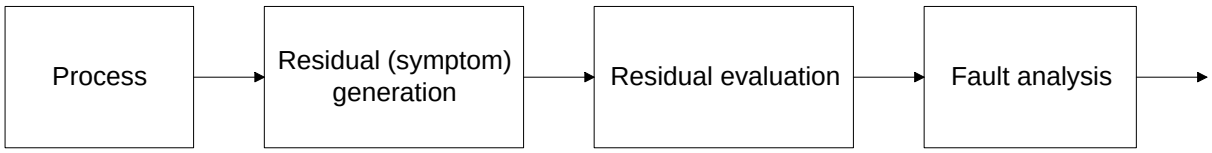


Fig. 2.9 - Schematic representation of the FDD procedure.

A Fault can be defined as a non-permitted deviation of at least one characteristic property of a variable from an acceptable behaviour (Appendix A). Therefore, the fault is a state that may lead to a malfunction or failure of the system. The time dependency of faults can be distinguished as shown in Fig. 2.10 (Isermann, 2004): a) abrupt fault (stepwise); b) incipient fault (drift-like); c) intermittent fault.

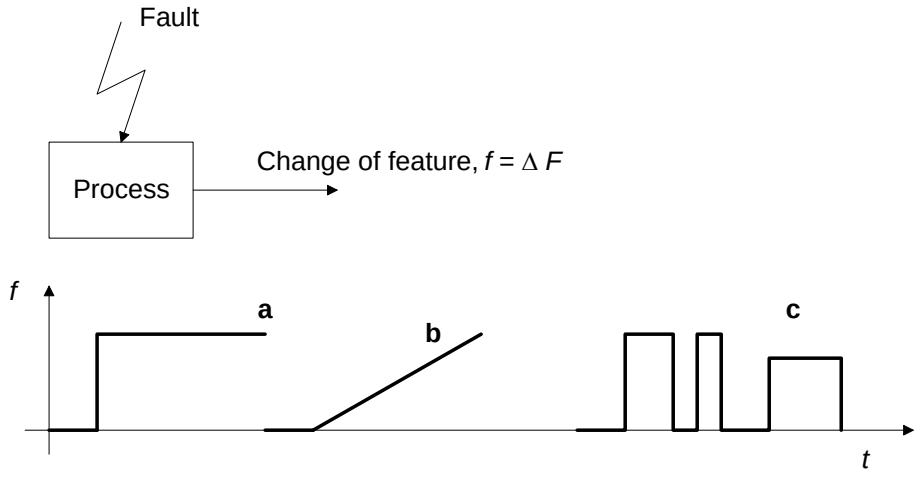


Fig. 2.10 - Time dependency of faults: a) abrupt; b) incipient; c) intermittent.

With regard to the process models, the faults can be further classified (Isermann, 2004). According to Fig. 2.11, additive faults influence a variable *y* by an addition of the fault *f*, and

multiplicative faults by the product of variable y with f. Additive faults appear as offsets on sensors and actuators, whereas multiplicative faults are parameter changes within a process.

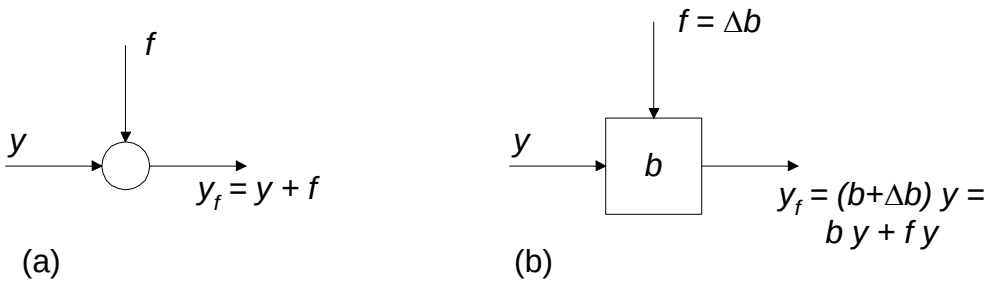


Fig. 2.11 - Basic models of faults: a) additive faults; b) multiplicative faults.

Some faults do not affect the system structure and are termed non-structural faults, and others affect the system structure and are designated structural faults (Blanke, et al., 2003).

For non-structural faults, the only changes they cause are in the mathematical expressions of the constraints. Parametric faults, in which only the values of the parameters are changed, are an example of non-structural faults. An example, is a change on a resistor from the nominal value R_n to a faulty value, $0 < R_f < \infty$. The occurrence of the fault just changes the constraint $u - R_n\, i = 0$ into the constraint $u - R_f\, i = 0$.

Structural faults change the set of the constraints and variables which are to be considered. For a resistor, an example of such faults is $R_f = 0$ (a complete short-circuit), changing the constraint into $u = 0$. Other examples of structural faults are blocked valves and leaks in water tank (liquid level) systems.

2.5.3 Fault Detection Methods

Fault Detection is the task of determination of the faults present in a system, and the time of detection, as described in Appendix A.

A number of different fault detection approaches making use of either hardware or software methods have been proposed over the years (Frank, et al., 1997; Isermann & Balle, 1997; Chen & Patton, 1999).

Depending on the method of features (residuals, etc) generation, the methods of fault detection can be divided into three main categories, as depicted in Fig. 2.12. The main sub-categories are also shown.

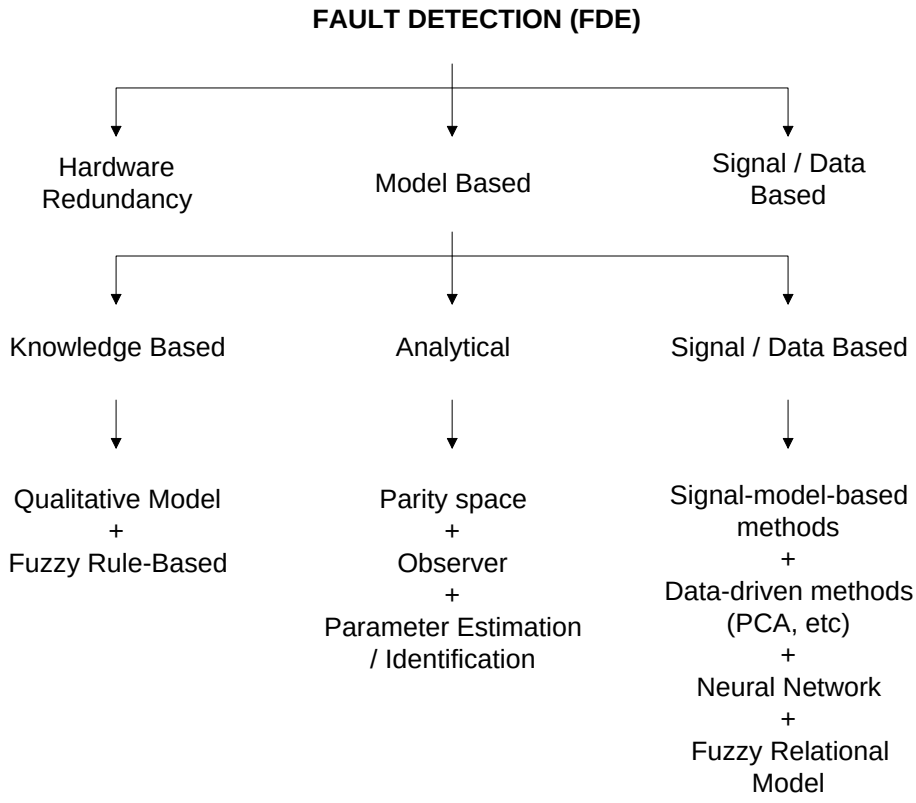


Fig. 2.12 - Scheme of features (residuals, etc) generation approaches.

A good overview of the historical development of model-based fault detection is presented in the paper (Isermann & Balle, 1997). The main development began at various places in the early 1970's. Beard (1971) and Jones (1973) reported an observer-based fault detection method for linear systems. The development of fault detection methods up to the respective times is summarized in the books Pau (1981), Chen & Patton (1999), and in survey papers by Gertler (1988), Frank (1990) and Isermann (1994).

The signal/data-based approaches to FDE that are not based on models are well established in practice. Typical symptoms are the magnitudes of the measured signals, quadratic mean values, limit values, trends, spectra power densities, correlation coefficients, covariances, etc. These signal-based approaches are limited in their efficiency, in particular for the detection of faults that occur in the dynamics of the system under investigation (Frank, 1996).
More powerful than the non model-based approaches are the model-based approaches to FDE. The most powerful fault detection (FDE) approaches are those based on a process model, where either analytical, data-based or knowledge-based models, or combinations of them can be used.

The basic idea behind the model-based fault detection approach is to use the nominal model of the system to generate features (residuals, etc) that contain information about the faults, as depicted in Fig. 2.13 (Isermann, 1997). The quality of the model is of crucial importance for both fault detectability and isolability, and also to guarantee a low rate of false alarms.

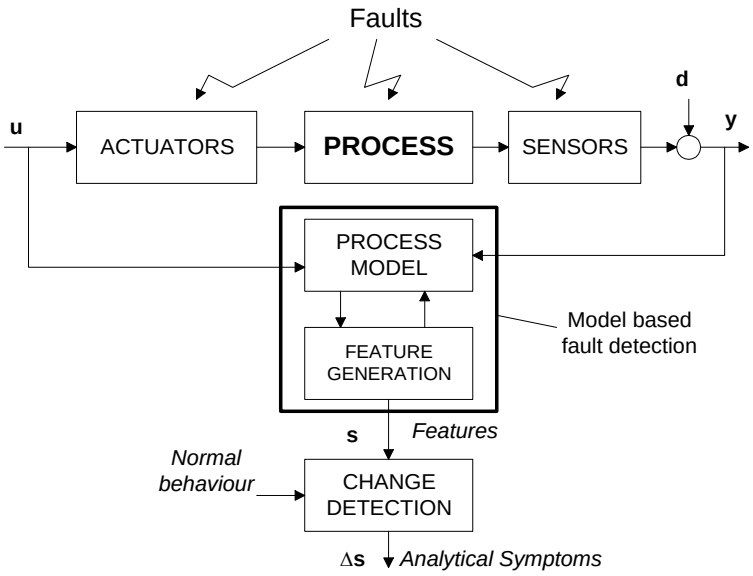


Fig. 2.13 - Scheme for model based fault detection.

If only output signals can be measured, Signal Model-Based Methods for FDE can be applied. In particular, vibrations, which are related to rotating machinery or electrical circuits, can be detected. Parametric signal models like ARMA type models, which allow the main frequencies and their amplitudes to be directly estimated (Isermann, 2004), can be used. Signal Model-Based Methods for FDE are essentially based on (Isermann & Balle, 1997): a) bandpass filters; b) spectral analysis (FFT); c) maximum-entropy estimation.

The Classical Analytical Model-Based Approaches (Parity Space, Observer, and Parameter Estimation / Identification) to residual generation deserves special attention due not only to their historical impact on the development of all the FDD techniques, but also to the impact on the applications developed. Clearly a perfect analytical model represents the deepest and most concise knowledge of the process. But in practice, precise analytical models are usually not or hardly ever available.

The Model-Based Data-Driven Approach for FDE is based directly on process data. In large-scale industries (chemical, manufacturing, nuclear, etc) which incorporate complex systems, the data-driven FDD methods are those most used, in order to avoid high cost analytical

models (Chiang, et al., 2001). The strength of data-driven techniques is their ability to transform the high-dimensional data into a lower dimension, in which the important information is captured, and can be interpreted by operators and engineers. The main drawback of data-driven measures is that the performance is highly dependent on the quality and quantity of the process data. The FDE approaches based on Neural Networks are extremely important especially when the systems are nonlinear (Chen & Patton, 1999).

The Knowledge-Based Approach for fault detection and process monitoring use knowledge based methods such as causal analysis (signed direct graphs, symptom trees), expert systems (knowledge base and inference engine) and pattern recognition (data patterns and fault classes) (Chiang, et al., 2001). These techniques are based on qualitative models, which can be obtained through causal modeling of the system, expert knowledge, fault-symptom examples, or a detailed description of the system.

Features (physical or model parameters, residuals, variances, etc) are generated in the fault detection task. The symptoms are the set of deviations of the features from the nominal case. The symptoms must reflect a fault univocally. In order to perform the isolation and the identification of a fault, the symptoms must be introduced in an intelligent decision making system (knowledge-based system) that maps the symptoms into the respective faults.
Since the early work of Beard (1971), system models common in control theory have been used for the design of FDE systems. It is well known that different types of applications require different types of models. Usually, for control system analysis and design, the system model has to represent the dynamic input-output behaviour of the system, and should be as simple as possible; in some cases, the model is drastically simplified and linearized ignoring many of the attributes of the physical nature of the system. Usually, for fault detection a representative model of high accuracy is needed, which is in general of higher complexity than the one for control. But under certain circumstances, models for FDE can also be simpler than those for control, which has often been overseen in the FDD society. The key point is that for FDE only that part of the model which reflects the faults of interest is needed and, with respect to robustness, is not or only weakly affected by disturbances and modeling uncertainty (Frank, et al., 1997).

Typically, input/output models in a form of difference equation or state-space models can be used to represent dynamic systems (linear or nonlinear, SISO or MIMO) in continuous-time or in discrete-time. A brief introduction to the subject is given here: Process Models and Fault

Modeling in Linear Systems. More information can be found in (Isermann, 2004; Isermann, 1995). Processes of SISO type with lumped parameters which can be linearized around one operating point can be described by a difference equation (ARX model) in discrete time:

$$y(k) + a_1\, y(k\text{-}1) + \ldots + a_{na}\, y(k\text{-}n_a) = b_1\, u(k\text{-}n_d) + \ldots + b_{nb}\, u(k\text{-}(n_d + n_b\text{-}1)) \qquad \text{Eq. 2.31}$$

The process model can be written in vector form:

$$y(k) = \varphi^T(k)\, \theta \qquad \text{Eq. 2.32}$$

with the parameter vector and the data vector given by, respectively, $\theta = [a_1\ a_2 \ldots\ a_{na}\ \ b_1 \ldots\ b_{nb}]^T$, $\varphi(k) = [-y(k\text{-}1)\ \text{-} \ldots \text{-}y(k\text{-}n_a)\ + u(k\text{-}n_d)\ + \ldots + u(k\text{-}(n_d + n_b\text{-}1))]^T$

Additive (offset) faults $f_u(k)$ and $f_y(k)$ at the input and the output signals can be modelled by:

$$y(k) = \varphi^T(k)\, \theta + \Delta\varphi^T(k)\, \theta + f_y(k) \qquad \text{Eq. 2.33}$$

$$\Delta\varphi(k) = [-y(k\text{-}1)\ \text{-} \ldots \text{-}y(k\text{-}n_a)\ + f_u(k\text{-}n_d)\ + \ldots + f_u(k\text{-}(n_d + n_b\text{-}1))]^T \qquad \text{Eq. 2.34}$$

For multiplicative (parametric) faults $\Delta\theta$, the fault model comes:

$$y(k) = \varphi^T(k)\, [\theta + \Delta\theta(k)] \qquad \text{Eq. 2.35}$$

A state-space representation for a MIMO linear process is

$$\begin{aligned} \mathbf{x}(k\text{+}1) &= \mathbf{A}\, \mathbf{x}(k) + \mathbf{B}\, \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\, \mathbf{x}(k) \end{aligned} \qquad \text{Eq. 2.36}$$

with p input signals $\mathbf{u}(k)$, and r output signals $\mathbf{y}(k)$. Offset changes $\mathbf{f}_L(k)$ of the states $\mathbf{x}(k)$ and $\mathbf{f}_M(k)$ of the output $\mathbf{y}(k)$ are then modelled by

$$\begin{aligned} \mathbf{x}(k\text{+}1) &= \mathbf{A}\, \mathbf{x}(k) + \mathbf{B}\, \mathbf{u}(k) + \mathbf{L}\, \mathbf{f}_L(k) \\ \mathbf{y}(k) &= \mathbf{C}\, \mathbf{x}(k) + \mathbf{M}\, \mathbf{f}_M(k) \end{aligned} \qquad \text{Eq. 2.37}$$

For parametric faults (**ΔA**, **ΔB**, **ΔC**), it holds

$$\mathbf{x}(k+1) = [\mathbf{A} + \Delta\mathbf{A}]\,\mathbf{x}(k) + [\mathbf{B} + \Delta\mathbf{B}]\,\mathbf{u}(k) \qquad \text{Eq. 2.38}$$
$$\mathbf{y}(k) = [\mathbf{C} + \Delta\mathbf{C}]\,\mathbf{x}(k)$$

A brief introduction of the main classical analytical model-based approaches to symptom (residual) generation in linear systems is given here. A more profound treatment of this subject can be found in (Isermann, 2004; Frank, et al., 1997; Chen & Patton, 1999; Gertler, 1998). The three most important analytical model based fault detection methods are: parity equations, state estimation (or observer), and parameter estimation / identification. Many people consider the Chow-Willsky scheme (Chow & Willsky, 1984) the parity relation (equations) approach.
For the case of linear systems, relationships exist between some of these FDE methods (Gertler, 2000). Parity equations (relations) are equivalent to observers, since any Luenberger residual generator may be implemented as a set of parity relations. PCA based FDD of additive (actuator and sensor) faults is linked to parity equations.

For the parity equations (or parity space) approach mainly two types of equations can be used to generate residuals: output error equation and polynomial error equation. In the *z*-domain, the output error equation is given by

$$R_e(z) = (G_P(z) - G_M(z))\,U(z) \qquad \text{Eq. 2.39}$$

with $G_M(z) = B_M(z) / A_M(z)$ being the discrete transfer function of a reference (fixed) model, and $G_P(z) = B_P(z) / A_P(z)$ is a model identified on-line. The polynomial error equation is given by

$$R_e(z) = A_M(z^{-1})\,Y(z) - B_M(z^{-1})\,U(z) \qquad \text{Eq. 2.40}$$

with $A_M(z^{-1}) = 1 + a_1\,z^{-1} + \ldots + a_{na}\,z^{-na}$ and $B_M(z^{-1}) = b_{nd}\,z^{-nd} + \ldots + b_{nd+nb-1}\,z^{-(nd+nb-1)}$.

For the state observer (or estimator) the generation of residuals can be carried out using one of the following equations. Changes of state estimates given by

$$\Delta\hat{\mathbf{x}}(k) = \mathbf{x}(k) - \mathbf{x}_0(k) \qquad \text{Eq. 2.41}$$

or output errors expressed by

$$\mathbf{r}_e(k) = \mathbf{y}(k) - \mathbf{C}\,\hat{\mathbf{x}}(k) \qquad \text{Eq. 2.42}$$

or filtered output errors written in the form

$$R_f(z) = W(z)\,R_e(z) \qquad \text{Eq. 2.43}$$

with $R_e(z) = TZ(\mathbf{r}_e(k))$ is the Z-transform of the output error $\mathbf{r}_e(k)$, and $W(z)$ represents a filter. It is also possible to use output predictor equations for FDE purposes.

The parameter estimation / identification approach usually generate residuals in two forms: changes of model parameter estimates or changes of process physical coefficients. The equation for the changes of model parameter estimates is given by

$$\Delta\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k) - \boldsymbol{\theta}_0 \qquad \text{Eq. 2.44}$$

and the changes of process physical coefficients is written in the form

$$\Delta\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k) - \mathbf{p}_0 \qquad \text{Eq. 2.45}$$

with $\hat{\mathbf{p}}(k) = f^{-1}(\hat{\boldsymbol{\theta}}(k))$.

Parity space and state estimation approaches have advantages for additive faults, and are therefore feasible for faults in sensors, actuators and in some cases for processes. For MIMO processes the analytical redundancy between the measured inputs and outputs increases; this is an advantage for the detection of sensor faults where the real input signal is unknown, and also for the detection of actuator faults if the actuator output is not measurable. Nevertheless, it is harder to obtain accurate process models with the cross-couplings for MIMO processes.
The parameter estimation / identification approach is especially suitable for multiplicative faults that change the dynamics of the process, actuators or sensors. It can also be used to deal

with additive faults at the input and output. Parameter Identification consists of the estimation of model parameters. The procedure of Parameter Estimation consists of a transformation of the mathematical parameters into the physical ones, which is often not unique and in many cases only feasible if the order of the model is low (Frank, et al., 2000a). A further problem with the application of the parameter estimation and adaptive observer approaches is the fact that the process always needs an input excitation verifying the persistent excitation conditions (PEC).

2.5.4 Fault Diagnosis Methods

Fault diagnosis is the task of determination of the kind, size, location and time of detection of a fault, as described in Appendix A. It follows fault detection, including fault isolation and identification (analysis). In most applications only the tasks of fault detection and isolation (FDI) are performed.

The fault diagnosis methods can be classified into two main categories (Isermann & Balle, 1997): classification methods, and reasoning methods. If several symptoms change differently for certain faults, one of the first ways of determining a fault is to use classification methods which indicate changes of symptom vectors. Some classification methods are: a) geometrical distance and probabilistic methods; b) artificial neural networks; c) fuzzy clustering. If more information about the relations between symptoms and faults is available in the form of diagnostic models, methods of reasoning can be applied. Diagnostic models can be expressed in the form of symptom-fault causalities. The causalities can be expressed as "if-then" rules. Then analytical as well as heuristic symptoms (from operators) can be processed. By considering them as inaccurate facts, probabilistic or fuzzy-set descriptions lead to a unified symptom representation. By forward and backward reasoning, probabilities or possibilities of faults are obtained as a result of diagnosis. Typical approximate reasoning methods are: a) probabilistic reasoning; b) possibilistic reasoning with fuzzy logic; c) reasoning with artificial neural networks.

Two main approaches are widely used for fault isolation: a) directional residuals, in response to a particular fault, the residual vector lies in a fault specific direction; b) structured residuals, each residual is sensitive to a subset of faults while insensitive to the rest. Diagonal residuals are a special case of both the directional and structured residuals, whereas each residual responds to a single fault. Directional residuals can be characterized by an influence matrix (Ono, et al., 1987; Doraiswami & Stevenson, 1996). Structured residuals are usually

characterized by a binary structure called the fault to residual incidence matrix or residual structure (Gertler, 1998). Residuals appear on the rows and faults on the columns.

Residuals are designed for the following requirements (Gertler, 2000): a) disturbance decoupling: residuals have to be insensitive to the disturbances while maintaining sensitivity with respect to (some) faults; b) isolation enhancement: residuals need to have special properties to support fault isolation; c) resilience to noise: the detection and isolation of faults from the residuals, even in the presence of (non-excessive) noise, should be possible.

A single residual may be sufficient to detect faults, but for fault isolation a set (vector) of residuals is usually required (Gertler, 2000). The enhancement of residuals can be obtained via two steps: a) first the computation of a primary residual set, and afterwards a transformation; b) the direct generation of enhanced residuals. Noise resilience can be achieved by low-pass filtering, and a threshold test.

Among the most important properties of a FDI system is not only the fact that it has to be sensitive to faults in order to detect incipient faults, but it also has to be robust with respect to the unknown inputs in order to avoid false alarms. The quality of fault detection can be assessed with the aid of the ratio of fault sensitivity to the frequency of false alarms. The quality of fault isolation is highly dependent upon the available information about the system.

2.5.5 Trends and Applications

Each process monitoring and FDD methodology has its strengths and limitations. The combination of different FDD schemes can result in better process monitoring performance for many applications (Isermann, 1997; Gertler, 1998; Chen & Patton, 1999; Chiang, et al., 2001; Palma, et al., 2005d).

As in most situations the model parameters are unknown, commonly the parameter estimation method is first applied. Typical combinations of FDE methods are (Isermann, 1995): a) parameter estimation to obtain the model, state estimation for fast change detection, and parameter estimation (on request) for deep fault diagnosis; b) parameter estimation to obtain the model, parity equations for change detection with fewer computations, and parameter estimation (on request) for deep fault diagnosis. The integration procedure depends on the process, the fault types and the allowable computational effort. In some cases, for example in rotating electrical machines, a good FDE performance can be obtained by integrating the process model based FDE methods with the signal model based FDE methods.

An FDD approach designed to provide satisfactory sensitivity to faults, associated with the necessary robustness with respect to modeling uncertainty, is called a robust FDD scheme

(Chen & Patton, 1999). The effect of modeling uncertainties is therefore the most crucial point in the model-based FDD concept, and the solution to this problem is the key for its practical applicability. When residuals cannot be made robust against system uncertainty, the robust FDD can be achieved by robust decision making; for example, a supervisor based on a fuzzy logic approach can be used for residual evaluation and decision-making. In a practical application, a situation where the conditions for a perfectly robust residual generation are met will rarely be found. One possible solution to increase the robustness of the FDD schemes is the combination of different approaches (analytical and data based, data based and knowledge based, etc). In most practical problems of fault detection and diagnosis a hybrid (combined) approach is required to guarantee a reasonable performance (Isermann, 2004; Palma, et al., 2005c). This hybrid approach combines different FDD methodologies.

Because of the many publications and increasing number of applications, it is interesting to show some trends. Peter Ballé has performed a literature study of Conference Preprints (Isermann & Balle, 1997) and presents some results. Only 165 contributions with applications were taken into account in the study: a) simulation of real processes (#55); b) full-scale industrial processes (#48); c) large scale pilot processes (#44); d) small-scale laboratory processes (#18). Mechanical and electrical processes, especially the DC motor, are those mostly investigated. Parameter estimation (PE) and observer-based (OB) methods are used in the majority of applications on this kind of processes, followed by parity space (PS) and combined methods. Neural nets (NN) are used less frequently. Thermal and chemical processes are also investigated less frequently, and the most used FDE methods are the model data-based methods, mainly the data-driven and neural networks approaches. Most nonlinear processes under investigation belong to the group of thermal and fluid dynamic processes.
OB and PE methods for fault detection methods are mostly applied; they are used on nearly 70% of all the applications considered. More than 50% of sensor faults are detected using observer-based methods, while PE, PS and combined methods play a less important role. The OB method is also the one mostly used for detection of actuator faults, and is followed by PE and NN methods. The detection of process faults is mostly carried out with PE methods. Some methods can be used for the detection of more than one fault class. Linear (or linearized) process models have been used much more than nonlinear models. On processes with nonlinear models, OB methods are the most applied, but PE and NN also play an important role. The number of nonlinear process applications using nonlinear models has clearly increased over the years.

The evaluation of fault diagnosis methods is more difficult because of a lack of data. The field of classification approaches, especially with neural networks and fuzzy logic, has steadily been growing. Rule-base reasoning methods, especially with fuzzy logic, are also increasing. Also evident is an increase on the application of neural networks both for residual generation and for classification.

2.6 Fault Tolerance

A brief introduction to the fault tolerance problem is presented here. More ideas are given in Chapter 5, with an application related to the reconfiguration problem of the three-tank benchmark (Heiming & Lunze, 1999). The Fault Tolerance (FT) problem is gaining more and more importance due to the requirements of the industrial modern control systems, since they are complex systems requiring a high level of automation, an increase of product quality, the reduction of costs, high reliability, high availability, and high security levels. From the point of view of control, the Fault Tolerant Control (FTC) problem belongs to the domain of complex systems, where it is necessary to incorporate multi-disciplinary information mainly from the control area and from the computational intelligent domain. A good book in the area of fault tolerant control is (Blanke, et al., 2003), and a good survey paper is (Patton, 1997).

Every system can be subject to faults, where a fault on a single component can have effects on the performance and availability of the system as a whole, or even cause a critical failure. Our modern society depends strongly upon the availability and correct operation of complex technological processes like manufacturing systems, energy production systems, telecommunications systems, etc. In the general sense, a Fault is something that changes the behaviour of a technological system in such a way that the system no longer satisfies its original purpose (Blanke, et al., 2003).

In order to avoid a decrease in system performance or damage to machines and humans, faults must be quickly found, and fast decisions must be taken in order to stop the propagation of their effects. One of the goals of the supervision and control systems is to make the overall system fault tolerant. From a systems-theoretic viewpoint, Fault Tolerant Control concerns the interaction between a given system (plant subject to a fault f) and a controller as depicted in Fig. 2.14 (Blanke, et al., 2003). In a fault tolerant system the term controller is used in an extended sense, including not only the usual feedback or feed-forward control law, but also the decision making layer that determines the control configuration. The decision layer

analyzes the plant behaviour in order to identify faults and changes the control law to guarantee that the closed-loop system falls within a region of acceptable performance.

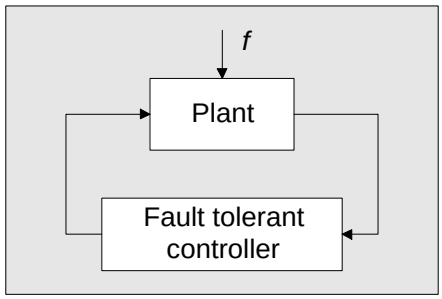


Fig. 2.14 - Fault tolerant system.

A fault tolerant controller has the ability to react to the existence of the fault by adjusting its activities to the faulty behaviour of the plant. In Fig. 2.14, the system is Fault Tolerant if it may be subjected to some fault *f*, but the fault effect is not perceptible to an external human observer, because the system remains available executing its purpose. Generally, the way to make a system fault tolerant consists of two steps that are carried out by a supervision system (Blanke, et al., 2003): 1) Fault Detection and Diagnosis, that is, any fault has to be detected and identified; 2) Control Re-Design, that is, the controller has to be adapted to the faulty situation so that the overall system continues to satisfy its goal. The supervision system must establish the control structure, and select the algorithm and parameters of the feedback controller.

Research into fault tolerant control has been largely motivated by the control problems encountered in critical systems like aircraft system design. In aircraft systems one of the main goals is to provide a self-repairing capability to enable the pilot to land the aircraft safely in the event of a serious fault (Patton, 1997). Interest has been stimulated mainly by two commercial aircraft accidents in the late 1970's. The paper written by Patton (1997) includes many references to application examples of FTC such as hazardous chemical plants, control of nuclear power plant reactors, space craft systems, aircraft systems, etc.

Typical applications that illustrate how the FTC methods can be applied under real practical conditions are described in the book (Blanke, et al., 2003), including different plant types: a three-tank system, a chemical process, a ship propulsion system and a steam generator.

2.7 Conclusions

The state of the art in the area of fault detection and diagnosis (FDD) has been presented in this chapter. A brief introduction to the fault tolerance problem was given.

An introduction to monitoring methods based on univariate and multivariate statistics was also presented. In processes where there is a redundancy or correlation between variables, it is advantageous to reduce the number of variables, maintaining an important quantity of original information. This dimensionality reduction can be achieved using linear or nonlinear Principal Components Analysis (PCA).

From a theoretical point of view a white-box model is more desirable to perform the FDD tasks, but in most cases it is very hard or even impossible to obtain. When the systems are complex, or hard to model, modelling based on black-box models (ARX models, NARX neural models, fuzzy models, etc) is usually a good and probably the only alternative. Taking a pragmatic view of models, the modeling should be guided by "usefulness" rather than "truth" (Ljung, 1999). A model for fault detection and diagnosis does not need to be equal to a model for control; it can be simpler, or more complex.

Typically different types of FDD approaches are more adequate to detect different kinds of faults. Model-based FDD methods based on parity equations and observers are most suitable to detect additive faults on actuators and sensors. The parameter estimation / identification approach is especially suitable for multiplicative (parametric) faults that change the dynamics of the process, actuators or sensors. The sliding window parameter estimation algorithms are most suitable to perform fault detection tasks than the recursive algorithms, since for a sliding data-window with length τ, it is known that the transient following a parameter jump lasts exactly $\tau - 1$ samples. The principal components regression (PCR) was formulated in this chapter as a theorem, and a sliding window algorithm SW-PCR was proposed.

Most of the efforts in the fault detection and diagnosis research area have been made on the development of methodologies based on linear models. A great challenge of recent years is the development of methodologies for nonlinear systems.

In most practical problems of fault detection and diagnosis a hybrid (combined) approach, incorporating different FDD approaches, is required to guarantee a reasonable performance. Nowadays, almost all complex systems incorporate basic fault detection modules. Advanced methods of supervision, fault detection and diagnosis, and fault tolerant control are needed.

3 Fault Detection and Diagnosis (FDD) Approaches for Linear Systems

Faults and failures can be good, if we learn with them (L. B. Palma).

3.1 Introduction

In Chapter 2, an introduction to fault detection and diagnosis (FDD) in dynamic systems was presented. In this chapter the focus is on FDD approaches for linear systems based on linear ARX black-box models, operating in closed-loop and with real-time constraints.

In the past, most of the research in model-based fault detection of dynamic systems has been done using analytical white-box models. However, a perfectly accurate and complete mathematical model of a physical system is never available; this is mainly true for large-scale complex industrial plants. This is one of the reasons why black-box models are gaining more and more interest and application (Ljung, 1999).

Even in linear systems, some faults can cause nonlinear effects. An example is a sensor fault that causes a saturation of the control signal, due to the controller action. In this situation, the linear models loose their validity.

Representative models used for FDD purposes do not need to be of full order, nor do they need to be equivalent to the ones used for control purposes. First, the type of fault to be detected must be defined, additive or multiplicative, and their location (on the sensors, on the actuators, or on the process). Then these elements must be used as a guideline to build signals and process models for FDD.

In this work, the main focus is on the detection and diagnosis of parametric (multiplicative) faults. Parameter estimation / identification methods need a process input excitation and are especially suitable for the detection of multiplicative faults (Isermann, 1997; Isermann, 2004).

In most practical problems of fault detection and diagnosis a hybrid (combined) approach is required to guarantee a reasonable performance (Isermann, 2004; Palma, et al., 2005c). This hybrid approach combines different FDD methodologies.

Most of the problems under study in this chapter (and in the dissertation) are restricted to single-input single-output (SISO) systems, but most of the proposed FDD methods can be extended to multi-input multi-output (MIMO) systems. The idea is to decompose the MIMO system in a set of multi-input single-output (MISO) systems. For this kind of MISO models, it is expected that the tasks of fault detection and isolation be simpler even when the aim is to detect more faults, since more models are available for the generation of features.

3.2 Fault Detection and Diagnosis based on ARX models

3.2.1 Introduction

In most practical problems white box models are not available, or they are hard and time consuming to obtain. Sometimes the only alternative is to use black-box models obtained via identification techniques. For the case of linear systems, the ARX model is a very popular black-box model (Ljung, 1999).

In this chapter some new FDD approaches based on ARX models are proposed. The main underlying idea is to detect and diagnose faults on the physical parameters of the plant via their symptoms reflected on the variation of ARX model parameters.

3.2.2 Problem Formulation

Typically, most of the dynamic systems in industry work under closed-loop control, as depicted in Fig. 3.1. This is the main reason why the focus of this dissertation is on FDD methods operating in closed-loop and in real-time.

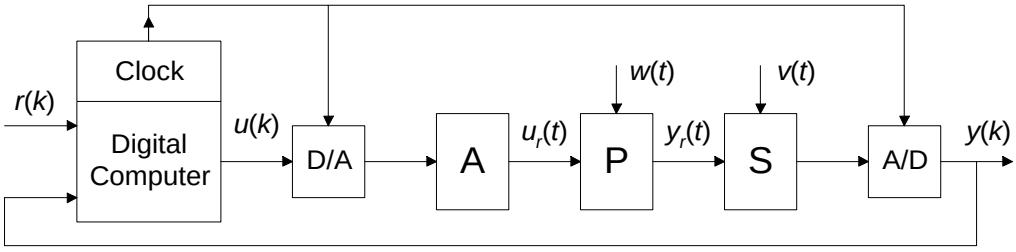


Fig. 3.1 - Closed-loop control architecture.

In Fig. 3.1, assuming a SISO system without loss of generality, the process plant is represented by the block P, and the blocks A and S represent, respectively, the actuator and the sensor. The blocks A/D and D/A are the analog-to-digital and the digital-to-analog converters. The digital computer implements the supervision, fault detection and diagnosis, and control algorithms. The digital signals are also represented in the figure: the reference signal, $r(k)$, and the input and output signals, $u(k)$ and $y(k)$, respectively. The other signals are analog signals: the real process input, $u_r(t)$, and the real process output, $y_r(t)$. The signals, $w(t)$ and $v(t)$, are, respectively, the disturbance input to the plant, and the disturbance or noise in the sensor.

For a single-input single-output (SISO) linear system, without loss of generality, the general problem under study in this chapter can be formulated as follows.

Problem 1. Given a continuous time LTI SISO dynamic system with unknown transfer function $G_0(s)$, find an on-line methodology for fault detection and diagnosis (FDD) in real-time operation. The main restrictions are: a) the process is modeled by a black-box ARX discrete time model M_x; b) the control architecture obeys the one depicted in Fig. 3.1; c) only the reference $r(k)$, the input $u(k)$ and the output $y(k)$ signals are measured; d) the FDD method must detect parametric (multiplicative) faults, and if possible also additive faults.

Assuming that a white box model is not available, it is necessary to perform system identification in order to obtain an acceptable low order ARX model. Different type of models relating the available signals in a system, under closed-loop (Fig. 3.1), can be used for fault detection and diagnosis. The most common models are: a) a model $M_{yu}(\boldsymbol{\theta})$, relating the output signal $y(k)$ and the input signal $u(k)$; b) a model $M_{yr}(\boldsymbol{\theta})$ relating the output signal $y(k)$ and the reference signal $r(k)$. If the aim is to detect only faults on the plant (sensors, actuators and process) then the model $M_{yu}(\boldsymbol{\theta})$ can be sufficient, but if faults on the controller need also to be detected then the model $M_{yr}(\boldsymbol{\theta})$ must be used.

3.2.3 Fault Modeling and FDD based on ARX Models

Assuming that a LTI SISO system is modeled by an input-output model $M_{yu}(\boldsymbol{\theta})$, then the model extended to include faults can be represented in the frequency domain by Eq. 3.1, where $G_u(z)$ is a known transfer function with state-space realization ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$), i.e., $G_u(z) = \mathbf{D} + \mathbf{C}\,(z\,\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{B}$. The term $\Delta Y(z)$ includes the disturbances and model uncertainty,

F is an unknown vector that represent all possible faults, and $G_f(z)$ is a known transfer function with state-space realization ($\mathbf{A}$, $\mathbf{E}_f$, $\mathbf{C}$, $\mathbf{F}_f$) (Frank, et al., 2000a):

$$Y(z) = G_u(z)\, U(z) + \Delta Y(z) + G_f(z)\, F(z)\ . \qquad \text{Eq. 3.1}$$

For each type of fault the transfer function assumes a different expression. For the sensor fault $G_f(z) = \mathbf{F}_f$, for the actuator fault $G_f(z) = G_u(z)$, and for the component fault $G_f(z) = \mathbf{C}\,(z\,\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{E}_f$. A great limitation in many real situations is that the transfer function $G_f(z)$ is usually unknown.

Considering that the residual signal $r_e(k)$ is expressed by the difference between the measured output value $y(k)$ and the predicted value $\hat{y}(k)$, Ding and Frank (1990) introduced the following construction of LTI residual generators, where $R_f(z)$ is a post-filter:

$$R_e(z) = R_f(z)\,(Y(z) - \hat{Y}(z))\ . \qquad \text{Eq. 3.2}$$

The FDI system design problem can be formulated as finding $R_f(z)$ so that the FDI system is stable and $R_e(z)$ is mostly sensitive to $F(z)$ and robust to $\Delta Y(z)$. In real situations, when few sensors are available and the goal is to detect many faults, find the filter $R_f(z)$ is usually a hard task.

Observing only the behaviour of the output signal $y(k)$ is not usually a good solution; it can be acceptable if combined with other fault detection techniques. The faults whose effects are not reflected persistently on the output signals are difficult to detect. In closed-loop, sometimes the controller masks the output effects of the faults. If we look only to an output signal, then mainly two approaches can be applied: a) a signal processing technique; b) a statistical data-based approach.

As mentioned earlier, the main goal of this work is to detect and diagnose parametric faults. Many authors argue that the parity equations and observers are most suitable for the detection of abrupt additive faults, and the parameter estimation / identification is more appropriate to detect parametric faults (Isermann, 1997; Gertler, 1998; Chen & Patton, 1999; Frank, et. al, 2000a).

Additive faults are usually detected using FDE approaches based on input-output behaviour $B_{u,y}$ (Blanke, et al., 2003). The pairs (**u**, **y**) are called input/output pairs (I/O pairs). The nominal behaviour $B_{u,y}$ of a plant is defined by the set of all possible pairs of trajectories **u** and **y** that may occur for the fault-free (faultless) case.

The detection of multiplicative faults is commonly performed by FDE methods based on on-line parameter estimation / identification. In this dissertation methods are proposed to detect and diagnose faults based on the parameter behaviour B_θ of ARX models, or features based on them. In fact, the ARX model parameters represent a "pseudo-state" of the process.

The great advantage of identification methods is that they can be used to detect both multiplicative and additive faults (Gertler, 1998). This can be understood in the following manner, following this example. For the case of Problem 1 under investigation, and observing Fig. 3.2, additive faults on actuator Δu or on sensor Δy, or a parametric fault $\Delta M_P(\boldsymbol{\theta}_P)$ on the plant, cause variations of the parameters associated with the ARX input-output model $M_{yu}(\boldsymbol{\theta})$ identified on-line.

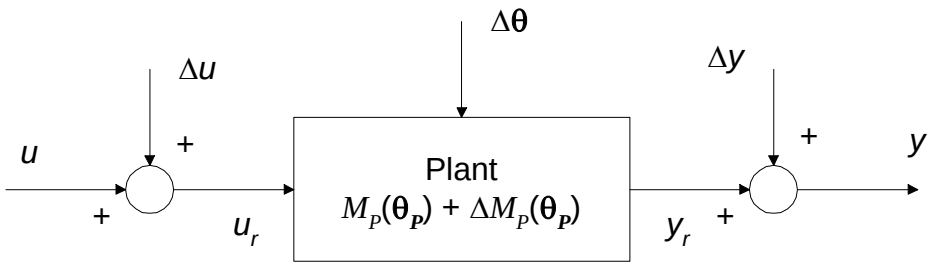


Fig. 3.2 - Typical faults on a SISO system.

For a LTI-SISO system, under nominal operating conditions (fault-free), the nominal ARX model in the predictor form can be expressed by

$$\hat{y}_0(k|\boldsymbol{\theta}_0) = \boldsymbol{\varphi}_0^{\mathrm{T}}(k)\,\boldsymbol{\theta}_0(k). \qquad \text{Eq. 3.3}$$

A typical fault (additive or multiplicative) usually causes a change in the system behaviour, B_θ. The system behaviour changes from fault-free $B_{\theta 0}$ to faulty $B_{\theta f}$. A fault usually causes a change in the regression vector from $\boldsymbol{\varphi}_0^{\mathrm{T}}(k)$ to $\boldsymbol{\varphi}_f^{\mathrm{T}}(k)$, and also a change in the parameter vector from $\boldsymbol{\theta}_0(k)$ to $\boldsymbol{\theta}_f(k)$. Consequently, the faulty system can be modeled by the faulty ARX model:

$$\hat{y}_f(k|\theta_f) = \varphi_f^T(k)\,\theta_f(k). \qquad \text{Eq. 3.4}$$

In the next sections new FDD methods for linear systems based on the parameter behaviour B_θ of the ARX model are proposed, requiring on-line parameter estimation / identification. The main idea is to detect and diagnose faults via changes on the ARX model parameters or on features based on them. Some of the FDD approaches proposed in this chapter can be extended to deal with nonlinear systems, since the on-line parameter estimation allows the adaptation to the system dynamic changes.

3.2.4 Closed-Loop Identification

Since most of the fault detection and diagnosis methods proposed in this work need on-line identification in closed-loop, a brief review is carried out here for a typical architecture depicted in Fig. 3.3.

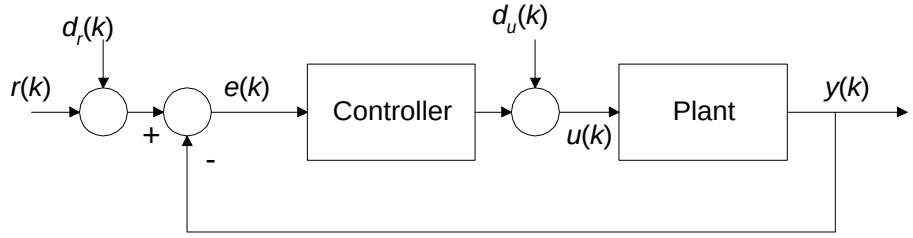


Fig. 3.3 - Closed-loop control, and dither signals.

In closed-loop identification, the data set must be informative. A data set is informative if it is capable of distinguishing between different models (Ljung, 1999). The controller must guarantee the persistence of input excitation, and an informative data set. In order to guarantee persistence of excitation, a dither signal must be added to the input signal, $d_u(k)$, or to the reference signal, $d_r(k)$, as depicted in Fig. 3.3. In this work, a white (Gaussian) noise with normal distribution (mean zero and variance σ^2) has been used as a dither signal added to the reference signal $r(k)$. Different approaches exist for closed-loop identification (Ljung, 1999): a) the direct approach; b) the indirect approach; c) the joint input-output approach.

In the direct approach the basic prediction-error identification method is applied, for example using the least-squares algorithm as a special case, or other derived algorithm. In identification of the ARX model $M_{yu}(\theta)$, the output signal, $y(k)$, and the input signal, $u(k)$, are used in the same way as for open-loop operation, ignoring any possible feedback, and not using the reference signal, $r(k)$.

The indirect approach consists in identifying the ARX model $M_{yr}(\boldsymbol{\theta})$ from reference signal, $r(k)$, to output signal, $y(k)$, and retrieve from that the input-output ARX model $M_{yu}(\boldsymbol{\theta})$, making use of the known controller.

For the joint input-output approach, the signals $y(k)$ and $u(k)$ are considered as outputs of a system driven by $r(k)$ and noise. From this joint model it is possible to recover the knowledge of the system ARX model $M_{yu}(\boldsymbol{\theta})$, and the controller.

The parameter estimation methods give the best model within the chosen model structure. The crucial question is: is this best model good enough? This is the problem of model validation. The question reveals several aspects (Ljung, 1999): a) does the model agree sufficiently well with the observed data?; b) is the model good enough for the purpose?; c) does the model describe the true system? The method to answer these questions is to confront the model with as much information about the true system as it is possible in practice.

Some typical methods for model validation are (Ljung, 1999; Soderstrom & Stoica, 1989): a) residual analysis (best fit criterion, statistical tests, etc); b) Akaike's information criterion (AIC); c) Akaike's final prediction error. In this dissertation, the best fit criterion and the Akaike's information criterion are used to validate the ARX model structures. The best fit criterion gives the percentage of the output y variation that is explained by the model, and is given by χ_{bfc}. Assuming that the norm is expressed by $\|.\|$, and the mean value of y by μ_y, then χ_{bfc} is expressed by

$$\chi_{bfc} = 100 \times \left(1 - \frac{\| \mathbf{y} - \hat{\mathbf{y}} \|}{\| \mathbf{y} - \mu_y \|}\right), \qquad \text{Eq. 3.5}$$

where $\mathbf{y}$ is the output signal, $\hat{\mathbf{y}}$ is the predicted model output, and μ_y is the output mean value.

Many plant faults are best characterized as changes in some plant parameters. Also, model errors resulting from shifting operating points may be described in terms of changes in model parameters. The FDI of parametric discrepancies by parameter estimation involves the identification of a reference model, in a situation when it is known (assumed) that no discrepancies are present, followed by repeated re-identification on-line. The residuals are obtained by comparison between the on-line estimates and the reference model.

When one (or more) of the plant parameters changes suddenly, a transient in the identified model parameters takes place. The estimate of the changed parameter does not follow the

change immediately. The sudden change of a single parameter causes transient errors in the estimate of the other parameters as well. This, of course, may be a major problem in the isolation of the change. To alleviate this problem, Gertler proposed a sliding window variant of the least squares algorithm (Gertler, 1998). In this dissertation a sliding window PCR parameter estimation algorithm is proposed; the equations and the pseudo-code of the algorithm can be found in section 2.3.7. If fault isolation is an objective, it is more advantageous to use a semi-batch algorithm for on-line parameter estimation, with a relatively short sliding data-window (Gertler, 1998).

With a sliding data-window of length τ, it is known that the transient following a parameter jump lasts exactly τ - 1 samples. Therefore, any isolation decision has to be delayed by τ - 1 samples following the detection of a change. The sliding window length τ must be selected according to the dominant process time constant.

3.3 FDD Approach using Dynamic Features of ARX Models

3.3.1 Introduction

A new fault detection and diagnosis (FDD) approach for linear systems based on dynamic features (static gain and bandwidth), computed from black-box ARX models, is proposed here. The aim of the work described here can be summarized in the following problem.

Problem 2. For a continuous time LTI SISO dynamic system with unknown transfer function $G_0(s)$, find an on-line methodology for fault detection and diagnosis (FDD) based on dynamic features (static gain and bandwidth) of ARX models. The main restrictions are: a) the process is modeled by a black-box ARX discrete time model M_x; b) the control architecture obeys the one depicted in Fig. 3.1; c) only the reference $r(k)$, the input $u(k)$ and the output $y(k)$ signals are measured; d) the FDD method must detect parametric (multiplicative) faults, and if possible also additive faults.

3.3.2 FDD Approach based on Dynamic Features of ARX Models

The static gain and the bandwidth of ARX models are the two main dynamic features considered here for fault detection and diagnosis purposes.

The static gain s_g of a process model can be estimated, each time instant k, according to the transfer function (Eq. 3.6) of an ARX model at frequency $z = 1$:

$$G_{yu}(\theta, z) = \frac{Y(z)}{U(z)} = \frac{b_1 z^{-nd} + \ldots + b_{nb} z^{-(nd+nb-1)}}{1 + a_1 z^{-1} + \ldots + a_{na} z^{-na}} \qquad \text{Eq. 3.6}$$

$$s_g(k) = G_{yu}(\theta, z = 1) = \frac{b_1(k) + \ldots + b_{nb}(k)}{1 + a_1(k) + \ldots + a_{na}(k)}$$

One important concept in system analysis is the bandwidth b_w of a LTI system. There are many different ways in which to define bandwidth. The bandwidth b_w, assuming a dynamic model with dominant poles of second order, can be approximately estimated on-line based on the estimated rise-time t_r (computed from the step response) of the ARX model (Oppenheim, et al., 1983):

$$b_w(k) \cong \frac{2\,\pi}{t_r(k)}\,. \qquad \text{Eq. 3.7}$$

Assuming that the faults on the physical parameters γ cause variations in the ARX model parameters θ, then each fault can be modeled by a different ARX model M_X. For each different ARX model a different set of features (patterns) $\{s_g\,,\, b_w\}$ is associated. Using the features s_g and b_w, a two dimensional features space is proposed in this work for fault detection and diagnosis as depicted in Fig. 3.4. The cluster F_0 is associated with the nominal operation, and the clusters F_i (i = 1, 2,…, f) are associated with the other faults.

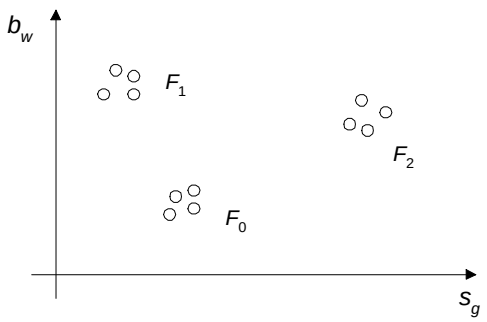


Fig. 3.4 - Features space based on static gain and bandwidth.

Here the fault detection (FDE) approach, the fault diagnosis (FDG) approach, and the combined FDD approach are presented. The general architecture is depicted in Fig. 3.5. Two types of ARX models can be used in this approach: a) an input-output ARX model M_{yu} relating the output signal $y(k)$ with the input signal $u(k)$, or a reference-output ARX model M_{yr} relating the output signal $y(k)$ with the reference signal $r(k)$. The sliding window principal components regression (SW-PCR) algorithm, described in section 2.3.7, is used on-line for estimation of model parameters.

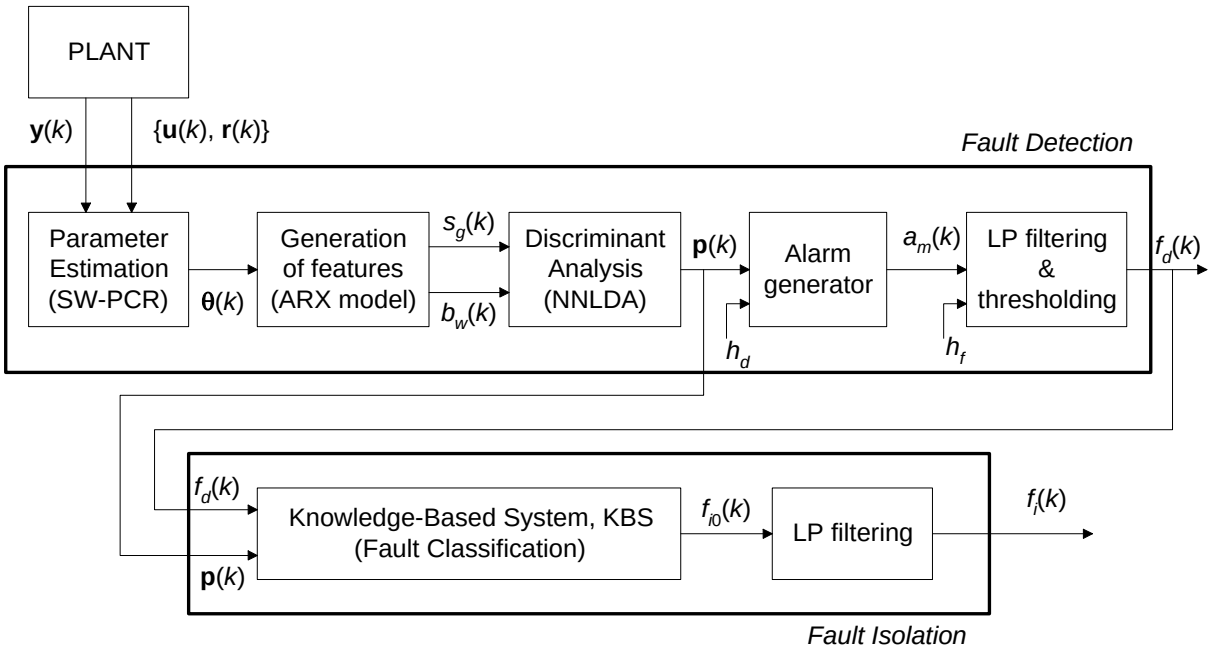


Fig. 3.5 - Architecture of the FDD approach based on dynamic features of ARX model.

The fault detection and isolation approaches proposed here use parameter estimation and a pattern classification approach using neural nonlinear discriminant analysis (NNLDA). NNLDA is used to discriminate between one of the patterns (clusters) associated with each fault. NNLDA makes it possible to define decision boundaries for each of the fault regions. In general the decision boundaries are nonlinear, and this is the reason why they are named hyper-surfaces. NNLDA can be viewed as a nonlinear approach that extends the statistical Fisher discriminant analysis (FDA) for the nonlinear cases. The discriminant analysis NNLDA enables the definition of decision boundaries needed for fault detection and isolation, and is more efficient than the geometrical techniques.

Fig. 3.6 depicts the architecture of the neural network (NN) used for nonlinear discriminant analysis (Asoh & Otsu, 1990). The NN acts as a pattern classifier, since it attributes a fault class to each data pattern. It is a feedforward multilayer perceptron neural network (FF-MLP-

NN) with 4 layers, $NN_{\{a\text{-}b\text{-}c\text{-}d\}}(\mathbf{W}_d, \ldots)$. The NN has an input layer (IL), a hidden layer (HL1) with hyperbolic tangent sigmoid (tansig) activation functions, and a hidden layer (HL2) and the output layer (OL) with linear activation functions (purelin). The neural network is trained off-line via the Levenberg-Marquardt optimization algorithm (Hagan, et al., 1995). If the number of inputs is n_i and the number of outputs is n_o, then the number of neurons in each of 4 layers is given by $[n_i\ m\ n_o\text{-}1\ n_o]$, and the value of m depends on the complexity of the classification problem (*e.g.*, $m = 10$).

In the training phase of the neural network, the input data vector **x** contains the data patterns (cluster data) associated with the features for each fault, and the output data vector **p** contains the class (pattern) associated. The length of the output vector is equal to the numbers of faults under study. Patterns with binary elements are used, *i.e.* for fault F_0 the fault class is 1 and is given by the output vector [1 0 0 0], and for fault F_1 the class 2 is given by the output vector [0 1 0 0], etc. Here, the neural model used for nonlinear discriminant analysis has the label M_{NNLDA}.

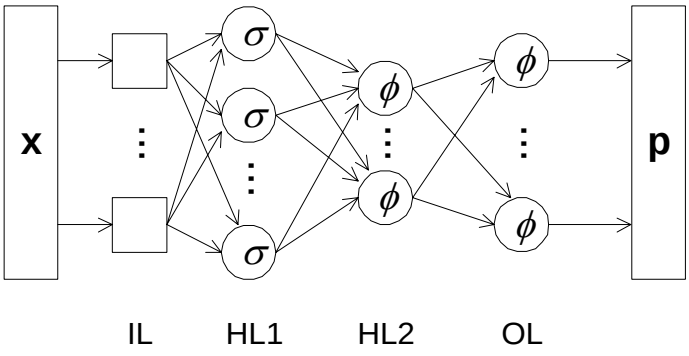


Fig. 3.6 - Architecture of the neural network for NNLDA.

Fault Detection Approach. The features static gain and bandwidth have been proposed for fault detection (FDE) in dynamic systems by Palma et al. (2002b). This FDE approach assumes that a fault in a plant can be detected on-line via changes on the estimated ARX model parameters, θ. This requires an on-line estimation of the ARX model parameters and the estimation of the dynamic features (static gain and bandwidth).

The basic principle for fault detection is described next. The nominal behaviour corresponds to the fault F_0, and is characterized by a data input pattern (cluster in a two dimensional graph) (Fig. 3.4). The on-line output (fault class) vector of the neural nonlinear discriminant analysis model M_{NNLDA} is expressed in the form $\mathbf{p} = [p_1\ \ p_2\ \ p_3\ \ p_4]$ for the case of 4 faults, i.e.,

the i^{th} position on vector **p** is denominated **p**(i) = p_i. For the case of nominal operation, corresponding to fault F_0, $p_1 \approx 1$ and $p_i|_{i\neq 1} \approx 0$.

A fault alarm signal is generated if the deviation from the nominal behaviour exceeds a certain threshold (h_d):, i.e., $p_1 < h_d$.

$$a_m(k) = 1 \Leftarrow p_1 < h_d \qquad \text{Eq. 3.8}$$

The threshold h_d is a design parameter. The threshold must be chosen in order to guarantee that the patterns (clusters in an n-dimensional space) for each fault are separated in the decision space of the neural network that implements the NNLDA. Typical values used here for the threshold h_d are around 0.9. To obtain a fault detection signal $f_d(k)$, the fault alarm signal $a_m(k)$ is low pass filtered. In the *z*-domain, assuming the *Z*-transform this is done by

$$F_d(z) = H_{lp}(z, \lambda)\, A_m(z) \qquad \text{Eq. 3.9}$$

The low pass filter $H_{lp}(z, \lambda)$ equations can be found in 2.4.4. Finally, the low pass filtered signal is compared to a threshold (a typical value is around 0.5):

$$f_d(k) = 1 \Leftarrow a_m(k) > h_f \qquad \text{Eq. 3.10}$$

Fault Diagnosis Approach. The task of fault isolation is executed after the task of fault detection. The isolation is based on a knowledge base system (KBS). The isolation is performed via the analysis of the output pattern (class), **p** = [p_1 p_2 p_3 p_4], generated by the neural nonlinear discriminant analysis model M_{NNLDA}. The isolation of fault number j is achieved:

$$f_{i0}(k) = j \Leftarrow (round(p_{j+1}) = 1) \text{ and } (round(p_i)|_{i\neq j+1} = 0) \qquad \text{Eq. 3.11}$$

where *round*(.) is the round function to nearest integer. The fault isolation signal is also low pass filtered ($H_{lp}(z, \lambda)$, section 2.4.4):

$$F_i(z) = H_{lp}(z, \lambda)\, F_{i0}(z) \qquad \text{Eq. 3.12}$$

Only the fault isolation has been described and implemented, but this approach can be used also for fault analysis (identification). According to the ideas explained, in the next sub-section the algorithm proposed is described and an example of application is presented.

Effective fault detectability and isolability is achieved if the input fault patterns (symptoms) are separated in a m-dimensional space, and also if the neural network model M_{NNLDA}, that implements the nonlinear discriminant analysis, is able to separate well the input fault patterns.

3.3.3 Algorithm and Example

The main ideas of the new FDD methodology proposed in this work have been explained in the last section. Afterwards, the FDD algorithm is described, and an example is given. Here it is assumed that an input-output ARX model M_{yu} relating the output signal $y(k)$ with the input signal $u(k)$ is used. A reference-output ARX model M_{yr}, relating the output signal $y(k)$ with the reference signal $r(k)$, can also be used.

Next the algorithm for implementation of the FDD approach applied to a SISO system is presented, based on the general architecture depicted in Fig. 3.5, assuming that the nominal behaviour is termed the fault F_0, and also the existence of n more faults.

Algorithm 2. Fault detection and diagnosis based on dynamic features of ARX models (FDD-DF-ARX).

In off-line operation, the following tasks must be executed:

a. For each fault F_i of the set $F = \{F_0, F_1, \ldots, F_n\}$, the features (static gain and bandwidth) must be estimated running the faulty system in closed-loop operation, and using the sliding window SW-PCR algorithm for estimation of the ARX model parameters $\boldsymbol{\theta}$. Each pattern (cluster in two dimensions) data associated with the respective fault must be saved.

b. Proceed with the training of the neural network $NN_{\{a\text{-}b\text{-}c\text{-}d\}}(\mathbf{W}_d, \ldots)$ that implements the neural nonlinear discriminant analysis (NNLDA), using the patterns (clusters) for all the faults F_i of the set $F = \{F_0, F_1, \ldots, F_n\}$. The Levenberg-Marquardt optimization algorithm is used for the neural network training. This discriminant neural model is expressed here by M_{NNLDA}.

c. Determine the thresholds and the low pass filters parameters, in order to obtain a desired trade-off between rate of false alarms, rate of missed fault detections, and detection and

isolation delays. Different sets of experimental nominal data must be used to compute and validate the thresholds and filters parameters.

Each time instant k, the following steps must be executed on-line:

1. Sample the process output signal $y(k)$.
2. Estimate the parameters $\boldsymbol{\theta}(k)$ of the input-output ARX model M_{yu}, based on past input-output data vectors, $\mathbf{y}(k)$ and $\mathbf{u}(k)$, using the SW-PCR parameter estimation algorithm.
3. Compute the dynamic features of the ARX model, i.e., the static gain $s_g(k)$ and the bandwidth $b_w(k)$.
4. Compute the output vector $\mathbf{p}(k)$ of the neural network M_{NNLDA} that implements the nonlinear discriminant analysis (NNLDA) approach.
5. Generate an alarm, $a_m(k) = 1$, if the first element of vector $\mathbf{p}(k)$ termed p_1 exceeds the threshold h_d , i.e., $p_1 < h_d$. A typical value for the threshold is around 0.9.
6. Compute the fault detection signal $f_d(k)$ by low pass filtering, $H_{lp}(z, \lambda)$, the fault alarm signal, and by thresholding. The thresholding is expressed by the rule: if $a_m(k) > h_f$ then $f_d(k) = 1$ else $f_d(k) = 0$. A typical value for the threshold is around 0.5.
7. If a fault is detected, i.e. $f_d(k) = 1$, then proceed to fault isolation. Using a knowledge based system (KBS) classify (map) each output vector pattern $\mathbf{p}(k)$ into the respective fault class F_i. The KBS system has been implemented using if-then rules, but fuzzy if-then rules or neural networks can also be used if a larger number of faults must be isolated. The signal $f_{i0}(k)$ is then obtained. Finally the fault isolation signal $f_i(k)$ is obtained by low pass filtering the signal $f_{i0}(k)$.

■

The role of the low pass filtering of signals is of crucial importance in FDD systems. One of the main goals of the low pass filtering is to decrease the rate of false alarms. The drawback is the increase of the detection and isolation delays.

Example 1. Fault detection and diagnosis (FDD) based on dynamic features of ARX models applied to a first order continuous time system.

In this example, the proposed FDD approach based on dynamic features of an ARX model is applied to a first order system according to Algorithm 2. The transfer function of the continuous time system is given by $G_0(s) = Y(s) / U(s) = K / (\tau s + 1)$, with nominal

parameters $K = 1$ and $\tau = 1$ s. It is assumed that the sensor signal of the plant has a low noise variance of 1×10^{-8} .

Here, an ARX($n_a = 2$, $n_b = 1$, $n_d = 1$) discrete time model is used for modelling the continuous time system $G_0(s)$, and also to compute the dynamic features: static gain s_g and bandwidth b_w. The discrete model ARX(2, 1, 1) can be expressed by the difference equation given by $y(k) = -a_1\, y(k\text{-}1) - a_2\, y(k\text{-}2) + b_1\, u(k\text{-}1) + e(k)$. For nominal operation, using a sampling period of $T_s = 0.11$ s, the mean values of the identified ARX model parameters are given by $[a_1\ a_2\ b_1] = [-8.9\times10^{-1}\ -4.1\times10^{-3}\ 1.0\times10^{-1}]$. A dither signal with variance 1×10^{-3} has been added to the reference signal in order to guarantee persistent excitation conditions.

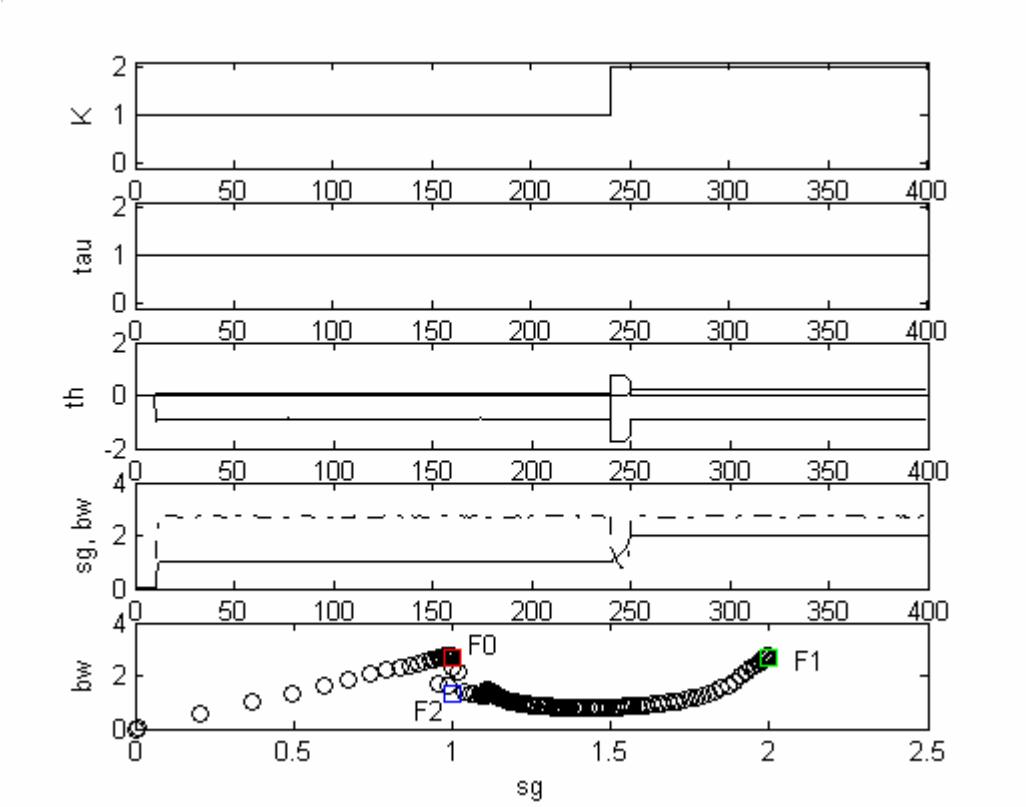


Fig. 3.7 - Dynamic features of ARX model for fault F1.

Fig. 3.7 shows the results obtained for a fault on the gain K of the continuous time model, for an experiment of 400 s. The gain K changes from a value of 1 to a value of 2. This corresponds to a transition from nominal operation (fault F_0) to fault F_1. From top to bottom, the signals can be observed: the gain K and time constant τ (tau) of the continuous time system, and the ARX model parameters ($\text{th} = \{a_1, a_2, b_1\}$) as a function of time. The penultimate graph shows the estimated static gain s_g (in solid line) and the estimated bandwidth b_w (in dash-dotted line). In the last graph a two dimensional graph for the

estimated features, static gain s_g and bandwidth b_w, can be observed. The two clusters (patterns) for faults F_0 and F_1 can be observed, the location of fault F_2 (blue square symbol), and also the trajectories associated with the transients behaviours. The fault F_2 is a changing in the time constant τ from 1 to 2, not shown in this experiment.

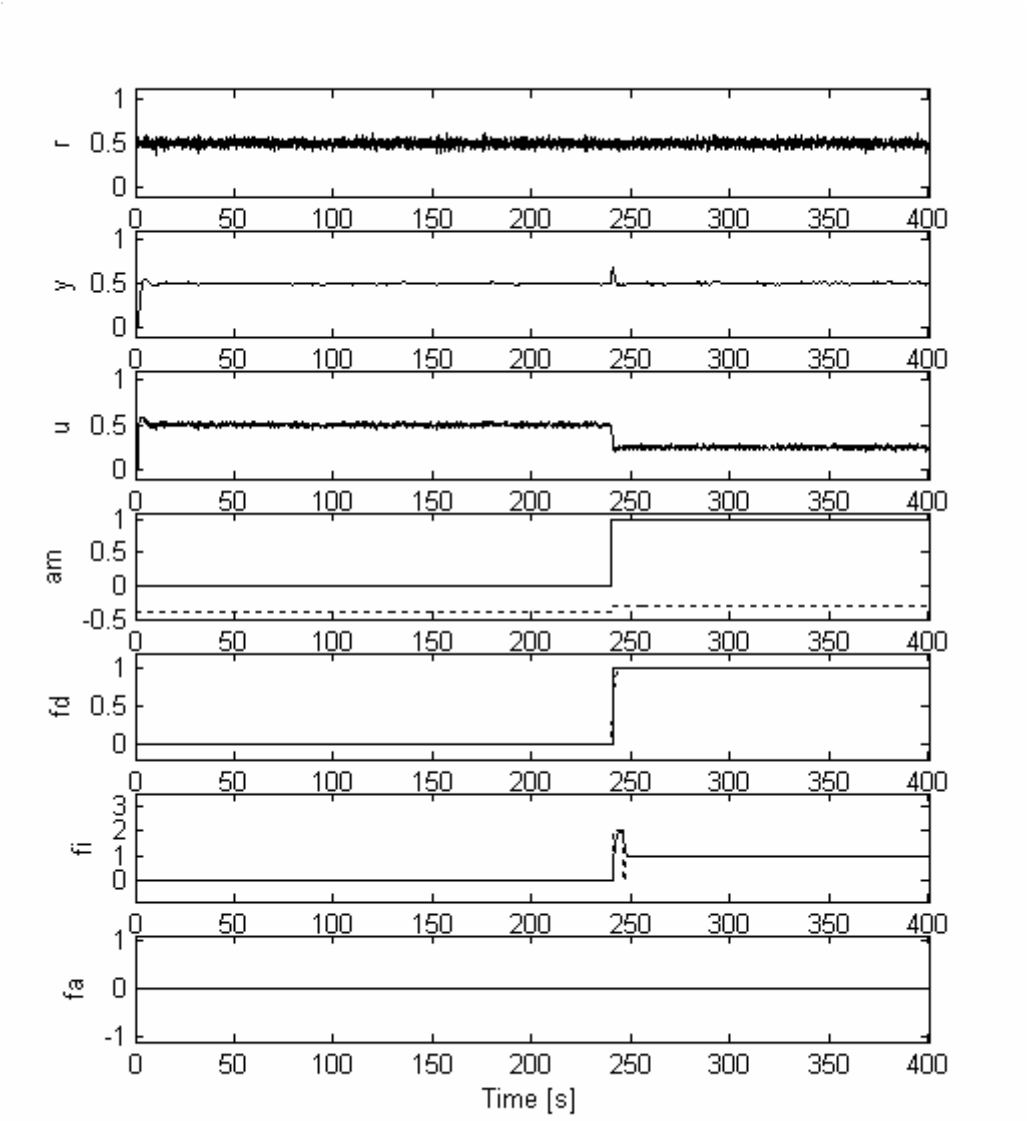


Fig. 3.8 - FDD signals for fault F1.

For this experiment, the control action comes from a digital linear PI controller, with $K_p = 0.4$ and $T_i = 0.4$ s, driving the system under closed-loop control (Astrom & Hagglund, 1988). Since here the parameters estimation is carried out using the sliding window SW-PCR algorithm, it is necessary to wait t_x seconds after fault detection in order to perform the fault isolation accurately. The time t_x is equivalent to the length of the sliding window of the SW-PCR algorithm. Here, t_x assumes the value of 10 seconds.

Fig. 3.8 shows, from top to bottom, the following signals: the reference signal $r(k)$, the output signal $y(k)$, the control input signal $u(k)$, the alarm signal $a_m(k)$, the fault detection signal $f_d(k)$, and the fault isolation signal $f_i(k)$. The fault analysis is not considered here. The fault F_1 occurs at time instant 240 s, and the detection delay is 0.8 s. The accurate fault isolation occurs around 10 s after fault occurrence. Theoretically, it is necessary to wait the time corresponding to the length of the sliding window of the SW-PCR parameter estimation algorithm; in this example, this length is 10 s. Both fault detection and isolation signals have been obtained by low pass filtering, $H_{lp}(z, \lambda)$, with the pole located at $\lambda = 0.9$. The pole location λ of the low pass filter is a design parameter, and must take into account the length of the sliding window of the parameter estimation algorithm.

In Example 2, an experiment is shown for different variances of sensor noise. The noise, an unpredictable disturbance, is particularly important in dynamic systems. The effect of increasing the variance of the sensor noise in the performance of the fault detection and diagnosis approach proposed is considered here. Similar noise effects also occur in other approaches proposed in this work, since most of them use on-line estimation of ARX models parameters.

The FDD approach depends on dynamic features (static gain and bandwidth) of ARX models that are computed based on on-line estimated model parameters. Increasing the variance of the sensor noise will cause an increase on the variance of the estimated parameters. For high parameter variances, a significant increase on the dispersion of the fault clusters (patterns) occurs, causing a degradation of the FDD performance. The static gain presents a small variance, while the bandwidth presents a significant increase on the variance. The sensibility of the bandwidth with respect to the variations of the ARX models parameters is higher since the bandwidth (estimated based on the rise-time) depends on the locations of the poles and zeros, and these locations are very sensitive to parameter variations. For high variances, the detection and isolation of small faults in some directions is not possible, since the nominal region increases with the increasing of the variance of sensor noise. Example 2 shows how the variance of the features used for FDD varies as a function of the variance of sensor noise.

Example 2. Effect of sensor noise in the model parameters and FDD features.

Tab. 3.1 shows the corresponding variances in the dynamic features of the FDD proposed approach for different values of variance of sensor noise.

This example is based on data captured for nominal operation, for the model $G_0(s) = Y(s) / U(s) = K / (\tau s + 1)$ described in Example 1, and for an ARX(2, 1, 1) model with parameters $[a_1\ a_2\ b_1] = [-8.9\times10^{-1}\ -4.1\times10^{-3}\ 1.0\times10^{-1}]$. Data has been captured under closed-loop control, using a PI controller with parameters $K_p = 0.4$ and $T_i = 0.4$ s, and a dither signal with variance 1×10^{-3} added to the reference signal.

The variance is denoted by $\sigma^2(.)$, and is computed for the ARX parameters and for the static gain and bandwidth. The great difference between the variance of the static gain and the variance of the bandwidth is clear.

For this FDD approach, when the variance of the sensor noise increases, it is necessary to compute a new nominal region larger than the old one, in order to avoid the increasing of the rate of false alarms. Another effect is the impossibility of detection and isolation of small faults in directions where the variance of the features is high. The use of adaptive thresholds can be considered to deal with different sensor noise situations.

Tab. 3.1 - Effect of sensor noise on the model parameters and FDD features.

Variance / Var. of sensor noise	$\sigma^2(a_1)$	$\sigma^2(a_2)$	$\sigma^2(b_1)$	$\sigma^2(s_g)$	$\sigma^2(b_w)$
1×10^{-8}	1.4×10^{-4}	1.2×10^{-4}	1.1×10^{-6}	7.0×10^{-10}	3.6×10^{-3}
1×10^{-6}	4.9×10^{-3}	4.5×10^{-3}	9.0×10^{-5}	1.2×10^{-7}	4.4×10^{-2}
1×10^{-4}	5.3×10^{-3}	5.9×10^{-3}	2.9×10^{-3}	6.1×10^{-6}	2.7

3.4 Fault Detection and Diagnosis based on the Influence Matrix (IMX) Method

3.4.1 Introduction

In typical real applications, it is not always possible to obtain residuals in the incidence matrix form that guarantee a strong isolation.

The influence matrix (IMX) method based on a geometrical approach, described in this section, is an alternative method for fault isolation. The IMX method assumes that a fault in a

feedback control system manifests itself as the deviation of the physical parameters (sensor gains, resistance, etc) from nominal values. As was explained in section 3.2, a reasonable way to achieve fault detection and diagnosis is based on the variations of the ARX parameters. If a relationship between physical parameters and ARX model parameters can be established, then it is possible to detect and diagnose faults in a plant. The input-output model usually depends on the physical parameters in a nonlinear way, whether the system is linear or nonlinear (Gertler, 2000). But if the plant operates in steady-state then an approximately linear relationship between model parameters and physical parameters can be assumed.

The main purposes of fault detection and diagnosis (FDD) of dynamic systems is to detect faults at the earliest stage as possible, before the performance of the system shows signs of marked abnormality, to locate the faulty part, and to estimate the degree of the fault. Many FDD methods have been investigated using various principles such as parity equations, observers, parameter estimation / identification, etc. The reason so many methods have been studied lies in the requirements for detection and diagnosis being so vast that it is usually difficult to satisfy all of these requirements by using only one FDD method. Sometimes, the alternative is to combine different methods in order to increase the performance (Isermann, 1997; Chen & Patton, 1999). Considering these circumstances and placing stress on practicality, Ono and co-workers tried to develop a method that has a rather simple algorithm and yet performs effectively (Ono, et al., 1987). In this dissertation, the method is called the Influence Matrix (IMX) method. The influence matrix is the Jacobian of the model parameter vector with respect to the physical process parameters.

Initially four themes have been studied by Ono and co-workers to be solved by the FDD IMX method. The first theme relates to the accuracy of the model of the system to be diagnosed. In most diagnostic methods, the Kalman filter is used to detect the error in the signal caused by the fault, and a statistical test is applied to detect and estimate the degree of the fault. In this case, if some differences exist between the actual system and the model used in the filter, there is a possibility of misjudgement. Therefore, the accuracy of the model is indispensable for accurate diagnosis. However, it is usually difficult to achieve an accurate modeling because of various errors that occur when modeling. Thus, a diagnostic method robust to model errors is desired.

The second theme concerns the method for fault detection and diagnosis. A method that is able to not only to detect the fault but also to indicate its location and to estimate its degree is desirable, in order to achieve fast repairs.

The third theme is about the type (mode) of the detectable faults. There are various types of faults, such as step changes, gradual changes, etc. A diagnostic system which can diagnose these various faults is desirable.

The last theme relates to computer programs. Taking into account that the detection and diagnosis is performed in real time, the algorithm should be as simple as possible in order to minimize the computer load.

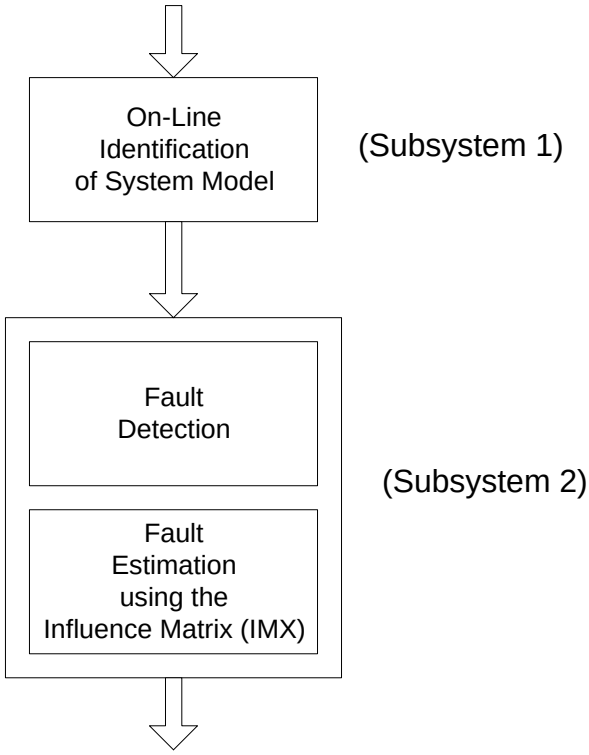


Fig. 3.9 - Architecture of the IMX FDD method.

Taking these matters into consideration, the FDD method proposed by Ono and co-workers consists basically of on-line parameter estimation / identification, fault detection and fault diagnosis using the influence matrix (IMX) method.

The FDD system consists of the following two subsystems as shown in Fig. 3.9 (Ono, et al., 1987). Subsystem 1 is used at the parameter identification phase to estimate the system model on-line, and in real time operation. Mainly, two types of ARX models can be used: input-output model $M_{yu}(\theta)$, or reference-output model $M_{yr}(\theta)$. Subsystem 2 works at the diagnostic phase to detect a fault by the deviation in the system model, and to estimate the degree of the deviation of the faulty physical parameter. In order to implement this approach it was chosen in this work to due successive identification and diagnosis, at each time instant *k*.

3.4.2 The IMX Method for Fault Detection and Diagnosis

The influence matrix (IMX) method enables the detection, isolation and estimation of faults, and has been developed for systems represented by input-output models (Ono, et al., 1987; Doraiswami & Stevenson, 1996; Posthan, et al., 1997). Assuming that the model parameters are multi-linear in the physical parameters (resistance, controller gain, etc), the influence of each physical parameter on the model parameters (called the influence vector) can be interpreted as a fault template (pattern) line associated with that physical parameter. The influence matrix is assumed to have been computed off-line, and stored for later use in on-line fault detection and diagnosis.

For FDD purposes, a low-order model, sometimes much lower than the true order, may be appropriate (Doraiswami & Stevenson, 1996; Posthan, et al., 1997). The identified model is merely a vehicle to estimate the faulty parameter, and not the true model of the system.

Considering the parameter (feature) vector of an ARX(n_a, n_b, n_d) model as a column vector $\boldsymbol{\theta} = [a_1\ a_2 \ldots a_{na}\ b_1 \ldots\ b_{nb}]^{\mathrm{T}}$, and the physical process parameters denoted by the vector $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2 \ldots \gamma_p]^{\mathrm{T}}$, the relationship between the model parameters and the physical parameters can be expressed as $\boldsymbol{\theta} = f(\boldsymbol{\gamma})$. Let the nominal values of the physical parameters be denoted by $\boldsymbol{\gamma}^{nom}$, and the vector of their deviations due to a fault by $\Delta\boldsymbol{\gamma} = [\Delta\gamma_1\ \Delta\gamma_2 \ldots \Delta\gamma_p]^{\mathrm{T}}$. The vector $\boldsymbol{\theta} = f(\boldsymbol{\gamma}^{nom} + \Delta\boldsymbol{\gamma})$ can be written in the form of a Taylor's expansion, and assuming there is only a single fault and each component of the feature vector is multi-linear in the physical parameters, then the Taylor's expansion will be simplified as (Posthan, et al., 1997):

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{nom} + \frac{\partial \boldsymbol{\theta}}{\partial \gamma_i}\Big|_{\gamma = \gamma^{nom}}\ \Delta\gamma_i \qquad \text{Eq. 3.13}$$

In (Eq. 3.13), the partial derivative of the feature vector, $\boldsymbol{\theta}$, with respect to the i^{th} physical parameter, γ_i, is termed the i^{th} influence vector, denoted by $\boldsymbol{\Omega}_i$. For each physical parameter, γ_j, the associated influence vector is defined as:

$$\boldsymbol{\Omega}_j^{\,nom} = \frac{\partial \boldsymbol{\theta}}{\partial \gamma_j}\Big|_{\gamma = \gamma^{nom}}\ ;\ \text{for } j = 1,\ldots, p\ . \qquad \text{Eq. 3.14}$$

The influence matrix (IMX), $\boldsymbol{\Omega}$, given by (Eq. 3.15) is the matrix whose columns are the p influence vectors $\boldsymbol{\Omega}_i$. The influence matrix is the Jacobian of the model parameter vector with

respect to the physical parameters, $\boldsymbol{\Omega} = \boldsymbol{\Omega}_{\theta,\gamma}$. The IMX matrix is computed off-line for the nominal operation (fault-free), and stored for later use in on-line FDD operation.

$$\boldsymbol{\Omega} = [\boldsymbol{\Omega}_1 \ldots \boldsymbol{\Omega}_p] = [\frac{\partial \boldsymbol{\theta}}{\partial \gamma_1} \cdots \frac{\partial \boldsymbol{\theta}}{\partial \gamma_p}]_{\gamma = \gamma^{nom}} \qquad \text{Eq. 3.15}$$

Lemma 1. Alignment between feature vector and influence vectors.
When a fault on a physical parameter $\Delta\gamma_i$ occurs, the feature vector deviation $\Delta\boldsymbol{\theta}$ will be aligned with the associated influence vector, $\boldsymbol{\Omega}_i$ (Ono, et al., 1987; Doraiswami & Stevenson, 1996; Posthan, et al., 1997):

$$\Delta\boldsymbol{\theta} = \boldsymbol{\Omega}_i \, \Delta\gamma_i \, . \qquad \text{Eq. 3.16}$$

■

Proof. Due to the fact that the feature vector is multi-linear in the physical parameters then the Taylor's expansion is simplified according to Eq. 3.13. This equation can be re-written in the form: $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^{nom} = \frac{\partial \boldsymbol{\theta}}{\partial \gamma_i}|_{\gamma = \gamma^{nom}} \Delta\gamma_i = \boldsymbol{\Omega}_i \, \Delta\gamma_i$.

■

The presence of noise will cause errors in estimation of ARX model parameters, $\boldsymbol{\theta}$, and the estimated $\Delta\boldsymbol{\theta}$ will never lie exactly in the direction of $\boldsymbol{\Omega}_i$, but will be expected to have the largest projection on $\boldsymbol{\Omega}_i$ than on the other influence vectors, assuming that the influence vectors are well separated in the Euclidean space of feature vectors.
After the faulty physical parameter, γ_i, has been determined, then an estimate of the value of its deviation, $\Delta\gamma_i$, may be obtained by solving a least-squares (LS) estimation problem. The optimal solution to this LS estimation problem is

$$\Delta\gamma_i^* = (\boldsymbol{\Omega}_i^T \boldsymbol{\Omega}_i)^{-1} \boldsymbol{\Omega}_i^T \Delta\boldsymbol{\theta} = (\boldsymbol{\Omega}_i^T \Delta\boldsymbol{\theta}) / \|\boldsymbol{\Omega}_i\|^2 \qquad \text{Eq. 3.17}$$

The magnitude of the projection of $\Delta\boldsymbol{\theta}$ on $\boldsymbol{\Omega}_i$ is given by Eq. 3.18, where $\hat{\boldsymbol{\Omega}}_i^T$ is the unitary vector in the direction of $\boldsymbol{\Omega}_i$, and $\phi(.)$ denotes the angle function between two vectors.

$$P_{\Omega_i}^{\Delta\theta} = \|\boldsymbol{\Omega}_i \, \Delta\gamma_i^* \| = |\hat{\boldsymbol{\Omega}}_i^T \, \Delta\boldsymbol{\theta}| = \|\Delta\boldsymbol{\theta}\| \cos(\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i)) \quad \text{Eq. 3.18}$$

According to the analytical geometry, for an *n*-dimensional space, the projection of a vector **u** on a direction defined by a vector **v** is given by:

$$proj_{\mathbf{v}}\mathbf{u} = \mathbf{u} \cdot (\mathbf{v} / \|\mathbf{v}\|) = \mathbf{u} \cdot \hat{\mathbf{v}} = \|\mathbf{u}\| \, \|\hat{\mathbf{v}}\| \cos(\phi(\mathbf{u}, \hat{\mathbf{v}})) = \|\mathbf{u}\| \cos(\phi(\mathbf{u}, \mathbf{v})) \quad \text{Eq. 3.19}$$

$$\|\mathbf{u}\| = \sqrt{\mathbf{u} \cdot \mathbf{u}} = \sqrt{u_1^2 + \ldots + u_n^2}$$

The following approach can be used for single fault isolation.

Lemma 2. Fault isolation based on largest projection.
Having estimated $\boldsymbol{\theta}$ (hence $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^{nom}$), the index of the faulty physical parameter, γ_i, is determined by the largest projection of $\Delta\boldsymbol{\theta}$ on the influence vectors (Ono, et al., 1987; Posthan, et al., 1997). That is

$$i = argmax_j \{|\hat{\boldsymbol{\Omega}}_j^T \, \Delta\boldsymbol{\theta}|\}; \, j = 1, \ldots, p \quad \text{Eq. 3.20}$$

■

Using the IMX concept and inspired by Lemma 2, a new fault isolation method (minimum angle criterion) is proposed in this work. The formulation is presented next in the form of a theorem.

Theorem 3. Fault isolation based on a minimum angle criterion for the IMX method.
Having estimated $\boldsymbol{\theta}$ (hence $\Delta\boldsymbol{\theta}$), the index of the faulty physical parameter, γ_i, is determined by the minimum angle (direct or indirect) between the feature vector deviation, $\Delta\boldsymbol{\theta}$, and the influence vectors, $\boldsymbol{\Omega}_i$ (Palma, et al., 2005c). Direct angles are angles between $\Delta\boldsymbol{\theta}$ and each $\boldsymbol{\Omega}_i$. Indirect angles are angles between $\Delta\boldsymbol{\theta}$ and each $-\boldsymbol{\Omega}_i$. The angle between two vectors is denoted here by the function $\phi(.)$. The faulty physical parameter is the one with the minimum angle obtained from all (direct and indirect) angles according to the expression

$$i = argmin_j \{\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_j); \, \phi(\Delta\boldsymbol{\theta}, -\boldsymbol{\Omega}_j)\} \; ; j = 1, \ldots, p \; . \quad \text{Eq. 3.21}$$

If $\Delta\boldsymbol{\theta}$ is more aligned with $\boldsymbol{\Omega}_i$ then an increase in the physical parameter γ_i occurs, i.e., $\Delta\gamma_i > 0$.
If $\Delta\boldsymbol{\theta}$ is more aligned with $-\boldsymbol{\Omega}_i$ then a decrease in γ_i occurs, i.e., $\Delta\gamma_i < 0$.

■

Proof. According to Lemma 1, when a fault $\Delta\gamma_i$ occurs $\Delta\boldsymbol{\theta} = \boldsymbol{\Omega}_i \Delta\gamma_i$. This means that $\Delta\boldsymbol{\theta}$ will be geometrically aligned in the direction of $\boldsymbol{\Omega}_i$. Then for a positive fault, $\Delta\gamma_i > 0$, the angle between vectors is $\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i) \cong 0$. For a negative fault, $\Delta\gamma_i < 0$, the angle is given by $\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i) \cong \pi \Rightarrow \phi(\Delta\boldsymbol{\theta}, -\boldsymbol{\Omega}_i) \cong 0$.
Lemma 2 can also be used to prove the theorem. Lemma 2 states that the isolated fault is the one with maximum projection, $P_{\Omega_i}^{\Delta\theta}$. The projection $P_{\Omega_i}^{\Delta\theta} = ||\Delta\boldsymbol{\theta}|| \cos(\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i))$ is maximum if $\cos(\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i))$ is maximum. If $\cos(\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i)) = 1 \Rightarrow \phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i) = 0$, and if $\cos(\phi(\Delta\boldsymbol{\theta}, \boldsymbol{\Omega}_i)) = -1 \Rightarrow \phi(\Delta\boldsymbol{\theta}, -\boldsymbol{\Omega}_i) = 0$.

■

The angle between two vectors, **u** and **v**, in a *n*-dimensional space can be computed by:

$$\phi(\mathbf{u}, \mathbf{v}) = acos((\mathbf{u} \cdot \mathbf{v}) / (||\mathbf{u}|| \, ||\mathbf{v}||)) \qquad \text{Eq. 3.22}$$

The influence matrix method described presents a methodology for fault isolation and analysis, but does not establish a method for fault detection.

A fault caused by a physical parameter change can be detected by a weighted sum of the squared of $\Delta\boldsymbol{\theta}(k) \in \Re^{m \times 1}$ (Ono, et al., 1987).
First, the parameter variation is computed, $\Delta\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k) - \boldsymbol{\theta}^{nom}$, where $\boldsymbol{\theta}^{nom}$ corresponds to a nominal reference model. The weighted sum squared of $\Delta\boldsymbol{\theta}(k) \in \Re^{m \times 1}$ can be expressed by $\nu(k)^2 = \Delta\boldsymbol{\theta}^T(k) \, \mathbf{W}_\theta \, \Delta\boldsymbol{\theta}(k)$, where $\mathbf{W}_\theta$ is a weighting matrix. The purpose of $\mathbf{W}_\theta$ is to give almost the same deviations, $\nu(k)^2$, to the minimum variations to be detected for every diagnosed parameter. Thus, almost the same fault detection performance can be obtained for all the parameters. By selecting the proper threshold value, η, the fault is detected as: $\nu(k) \geq \eta \;\Rightarrow\;$ fault at $k = k_f$.

Afterwards, an algorithm for fault diagnosis (isolation and analysis) based on the influence matrix is presented. Here an input-output ARX model M_{yu} relating the output signal $y(k)$ and the input signal $u(k)$ is assumed, but reference-output models of the type M_{yr} relating the output signal $y(k)$ and the reference signal $r(k)$ can also be used.

Algorithm 3. Fault diagnosis (isolation and analysis) based on the influence matrix (FDG-IMX).

In off-line operation, the following tasks must be executed:

a. For each fault F_i of the set $F = \{F_0, F_1, \ldots, F_n\}$ estimate the ARX model parameters and save the information.

b. For the faults of the set $F_f = \{F_1, \ldots, F_n\}$ compute the influence matrix $\boldsymbol{\Omega}$, i.e., the Jacobian of the model parameter vector with respect to the physical parameters, $\boldsymbol{\Omega} = \boldsymbol{\Omega}_{\theta, \gamma}$.

Each time instant k, the following steps must be executed on-line:

1. Sample the process output signal $y(k)$.
2. Estimate the vector of ARX model parameters, $\boldsymbol{\theta}(k)$, using the sliding window SW-PCR algorithm.
3. Compute the deviation of the on-line parameter vector relative to the nominal parameter vector, $\Delta\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k) - \boldsymbol{\theta}^{nom}$.
4. Compute direct and indirect angles between $\Delta\boldsymbol{\theta}(k)$ and each influence vector $\boldsymbol{\Omega}_j$.
5. Compute the fault isolation signal, $f_i(k)$, according to the minimum angle criterion formulated in Theorem 2. The fault isolated is given by $f_i(k) = i$, where i is expressed by:

$$i = argmin_j \{\phi(\Delta\boldsymbol{\theta}(k), \boldsymbol{\Omega}_j); \phi(\Delta\boldsymbol{\theta}(k), -\boldsymbol{\Omega}_j)\} \; ; j = 1, \ldots, p \quad . \qquad \text{Eq. 3.23}$$

6. Compute the fault analysis signal, $f_a(k)$, i.e. the fault magnitude according to the expression:

$$P_{\Omega_j}^{\Delta\theta} = \|\boldsymbol{\Omega}_j \, \Delta\gamma_j^* \| = |\hat{\boldsymbol{\Omega}}_j^T \, \Delta\boldsymbol{\theta}(k) | = \|\Delta\boldsymbol{\theta}(k)\| \cos(\phi(\Delta\boldsymbol{\theta}(k), \boldsymbol{\Omega}_j)) \qquad \text{Eq. 3.24}$$

■

An example of influence vectors in a two dimensional space is presented below.

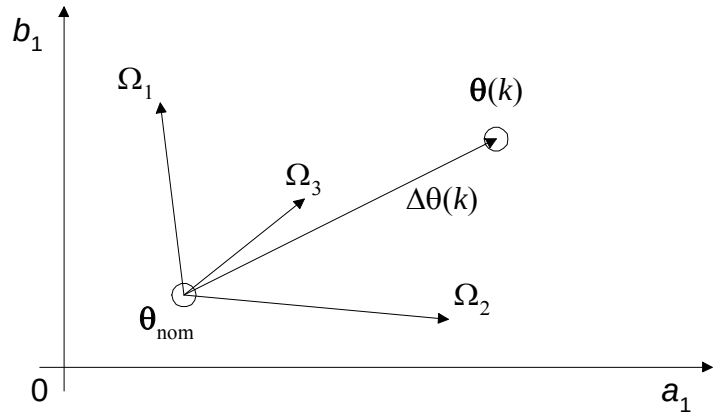


Fig. 3.10 - Influence vectors in two dimensional features space.

Example 3. Influence vectors in two dimensional features space.

If a continuous time system with transfer function $G_0(s)$ is to be modelled in discrete time by an ARX($n_a = 1$, $n_b = 1$, $n_d = 1$) model, then it is possible to obtain a features space in two dimension, given by the ARX model parameters $\{a_1, b_1\}$. For this ARX model the transfer function is given by $G_0(z) = b_1 z^{-1} / (1+ a_1 z^{-1})$. In on-line operation, the parameter vector is estimated in real-time and expressed by $\boldsymbol{\theta}(k) = [a_1(k)\ b_1(k)]$. Fig. 3.10 shows a typical scenario for the space of feature vectors, with 3 influence vectors, $\boldsymbol{\Omega}_i$, each one associated with a different possible fault. For this situation, the isolated fault is the fault F_3, according to the minimum angle criterion (Theorem 3). Since the vector $\Delta\boldsymbol{\theta}(k)$ is more aligned with the vector $\boldsymbol{\Omega}_3$ then an increase in the physical parameter γ_3 occurs.

The computation of the influence matrix (IMX) is done in off-line operation. Here, each of the physical parameters is perturbed from its nominal value, one-at-a-time, for a certain number of experiments (five, or more). The system is excited by a rich input signal, and the input-output signals are recorded. Then a system identification algorithm is employed, assuming a certain ARX model structure, for each of the input-output data records, corresponding to each of the experiments. Finally, the estimated feature vector (model parameters) are plotted against the physical parameters, and the slopes of the fitted lines to these curves, corresponding to the elements of the estimated influence matrix, are computed for each curve.

Instead of using all the ARX parameters $\boldsymbol{\theta} \in \Re^m$ for fault diagnosis, it is possible to use only a subset of parameters $\boldsymbol{\theta}_p \subset \boldsymbol{\theta}$, with $\boldsymbol{\theta}_p \in \Re^p$ and $p < m$. This subset $\boldsymbol{\theta}_p$ must include the parameters which are more sensitive to the faults, i.e., the ones with high slopes in the influence matrix.

The influence matrix method can also be used to diagnose multiple faults, by considering each combination of simultaneous faults as a new fault template.
The IMX method assumes a certain linear model for the system; hence it may be repeated for different orders. If the order of the system model is increased, then the deviations of the estimated parameters from linearity also increase. The assumption of linearity between physical parameters and the feature vector elements (model parameters) is usually valid in the continuous time domain and, in general, does not hold in the discrete time domain.

The IMX method can be also used to infer the true order of the system, and to controller tuning (Poshtan & Doraiswami, 1994). Estimated parameters of models with orders lower than the true order of the system were shown to better preserve multi-linearity with respect to the physical parameters, and hence may be more effectively used for fault detection and diagnosis purposes. Models with orders equal or greater than the true order are shown to be more appropriate for controller tuning (performance improvement). A good model is, therefore, application-dependent and may not necessarily be the true model.

Here a frequency interpretation for the FDD influence matrix method is given. The location of ARX model parameters in a n_x dimensional parameter space, P_θ, corresponds to locations of poles and zeros in the plane Z_p. In fact, each fault is characterized by a set of model parameters, $\theta_f = [a_1\ a_2\ \ldots\ a_{na}\ b_1\ b_2\ \ldots\ b_{nb}]$, and consequently certain poles and zeros located in the plane Z_p.
In this section, an approach for FDD based on the influence matrix (IMX) was presented. This method requires the off-line computation of an influence matrix. The IMX is the Jacobian of the model parameters with respect to the physical parameters. The method was explained assuming the existence of a single fault, but the IMX method can be extended to deal with simultaneous faults. The main idea of the FDD IMX method is based on variations of the model parameters. A fault changes the nominal behaviour. When a fault occurs, the model parameters move to another region in a $\Re^{n_x}$ dimensional space, assuming an ARX(n_a, n_b, n_d) model with $n_x = n_a + n_b$ parameters. The IMX method presents a reasonable performance in terms of geometric (directional) fault isolation, but for fault detection it does not establish a way of defining a threshold region. In this work, this problem has conducted the research into a direction of finding a method for fault detection using a threshold based on statistical properties of the ARX parameters, as will be described in the next section. This fault detection method will be later combined with the influence matrix approach.

3.5 Fault Detection Approach based on PCA applied to ARX Model Parameters

3.5.1 Introduction

The data-driven approaches to fault detection and diagnosis (FDD) are based directly on process data, and their strength lies in the ability to transform the high dimensional data into a lower dimension, in which the important information is captured. Principal Component Analysis (PCA) is one of the most widely statistical multivariate techniques used in industry, especially in large-scale systems that produce a large amount of multivariate data (Jackson, 2003; Jolliffe, 2002).

In most applications, PCA is applied off-line using a data matrix from a batch process as in chemical reactors in pharmaceutical processes (Lopes, 2001), or from chemical process simulators (Chiang, et al., 2001), or in atmospheric science (Jolliffe, 2002). There have been thousands of applications of PCA over the years, and some of them are mentioned in the book by Jackson (2003): psychology and education, quality control and fault detection, economics and market research, anatomy and biology, agriculture, etc. Few works exist related to dynamic PCA (dPCA) with applications operating on-line, and most of them have been applied to simulated models. Wang and co-workers (2005) present a dPCA approach applied to a model of a fluid catalytic cracking unit, and Klancar (2000) shows a dPCA method applied to a simulated laboratory three-tank pilot plant. In this work, an FDD approach based on PCA is proposed to work in real-time operation.

The PCA monitoring methods discussed previously in Chapter 2 assume implicitly that the observations at one time instant are statistically independent to observations at past time instances. For typical industrial processes, this assumption is valid only for long sampling periods. PCA can be used to take into account the serial (temporal) correlations by augmenting each observation vector with the previous h observations (Chiang, et al., 2001). A simple method to check whether correlations are present in the data is through the use of an autocorrelation chart of the principal components (Wei, 1990). If significant autocorrelation is present in the autocorrelation chart, the following approaches can be used: a) one approach is

to average the measurements over a data window; b) another approach is to incorporate cumulative sum or exponentially-weighted moving average charts (Montgomery, 1991).
In this work a new method for fault detection based on the application of a PCA approach to the ARX model parameters estimated on-line (Palma, 2005c) is proposed. This is a model-based FDD approach that combines an analytical approach (parameter estimation / identification) and a data based (statistical) approach based on principal components analysis.

3.5.2 Fault Detection Approach based on PCA

In the literature, the principal components analysis (PCA) method has been mostly applied directly to the input-output data (Chiang, et al., 2001; Jackson, 2003). But for some faults, the controller hides the fault effects and this render the FDI tasks difficult. In many faulty cases, the FDI based directly on model parameters can present a better performance than when based directly on the input/output data.
Inspired by the idea of fault detection and diagnosis based on the variations of the ARX model parameters in a LTI system, in this work a new method for fault detection based on PCA (Palma, et al., 2005c) is proposed. The main idea is to detect faults in a reduced space of the original ARX(n_a, n_b, n_d) parameter space. If the dimension of the original parameter space is $m = n_a + n_b$, then applying the PCA method to this space allows a dimension reduction to $a < m$ principal components. This space reduction allows, typically, a reduction of dimension from five or more, to two or three dimensions, allowing a better visualization and understanding of the system behaviour. The PCA decomposition allows the extraction of the most relevant information from the model parameters, since the sensitivity of each model parameter relative to a change on the physical parameters is different. Theoretically PCA guarantees uncorrelated principal components (PC's), and assuming the PC's obey approximately a normal distribution, thresholds for fault detection can be defined based on statistics assuming a normal distribution.
The application of PCA to all ARX parameters, $[a_1\ a_2\ \ldots\ a_{na}\ b_1\ \ldots\ b_{nb}]$, allows the extraction of the most significant variability in all the parameters, and also the projection of the scores in a two or three dimensional space (Palma, et al., 2005c). Applying the linear PCA technique only the linear relations between data are captured.

First, the PCA technique is explained through the construction of a PCA nominal model for the nominal region (without faults). This model will be used as a PCA reference model for

on-line fault detection. The on-line data will be projected onto the reduced space using this PCA reference model.

The PCA technique will be explained in a concise form, applied to ARX model parameters. For a system modelled by an ARX(n_a, n_b, n_d) model, with $m = n_a + n_b$ elements in the parameter vector, $\boldsymbol{\theta}(k) = [a_1(k)\ a_2(k)\ \ldots\ a_{na}(k)\ b_1(k)\ b_2(k)\ \ldots\ b_{nb}(k)]$, if n samples of the signals are acquired in the nominal region, then a data matrix $\mathbf{X} \in \Re^{n \times m}$ will be available to compute the PCA nominal model,

$$\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_m];\ \mathbf{x}_i \in \Re^{n \times 1}\ . \qquad \text{Eq. 3.25}$$

First the original data ($d_i = x_i$) is auto scaled via a standardization (transformation), in order to guarantee a zero mean and unity standard deviation, for each variable (each ARX parameter). This is done by

$$x_i = (d_i - \mu_i) / \sigma_i\ . \qquad \text{Eq. 3.26}$$

In the second step, the correlation matrix, $\mathbf{S} \in \Re^{m \times m}$, is computed according to

$$\mathbf{S} = \frac{1}{n\text{-}1}\,\mathbf{X}^{\mathrm{T}}\,\mathbf{X} = \mathbf{V}\,\boldsymbol{\Lambda}\,\mathbf{V}^{\mathrm{T}}\ . \qquad \text{Eq. 3.27}$$

The eigenvalue matrix $\boldsymbol{\Lambda} \in \Re^{m \times m}$, and the eigenvector matrix $\mathbf{V} \in \Re^{m \times m}$, are obtained by a singular value decomposition (SVD).

The third step consists in choosing the number of principal components, a, to retain in the PCA model. A number of techniques are available, but none appears to be dominant. In this work, the percent variance test is used (Chiang, et al., 2001; Jackson, 2003). This method determines a by calculating the smallest number of loading vectors (principal components) needed to explain a specific minimum percentage of the total variance (generally, greater than 80 %). The first a eigenvalues, $\lambda_1 \geq \ldots \geq \lambda_a$, explains the greatest variance, and the corresponding loading vectors define the scores space.

In the fourth step, the scaled data is projected onto the scores space. Selecting the columns of the loading matrix $\mathbf{P}_a \in \Re^{m \times a}$ to correspond to the loading vectors $\mathbf{V} \in \Re^{m \times m}$ associated with the first a eigenvalues, the projections of the observations $\mathbf{X} \in \Re^{n \times m}$ into the lower-dimensional space are contained in the scores matrix, $\mathbf{T}_a \in \Re^{n \times a}$, given by

$$\mathbf{T}_a = \mathbf{X}\,\mathbf{P}_a \,. \qquad \text{Eq. 3.28}$$

If the number of principal components is two, $a = 2$, then a two dimensional scores plot can be used to visualize the scores. In this scores space, the scores matrix for a window of length n can be represented in the form

$$\mathbf{T}_a = [\mathbf{t}_1\ \mathbf{t}_2],\ \mathbf{t}_i \in \Re^{n\times 1},\ i = 1, 2 \,. \qquad \text{Eq. 3.29}$$

Each line of $\mathbf{T}_a$ is a score. Each score is a projection of the original data in a reduced scores space. The score with coordinates (t_1, t_2) is represented by a point in a two dimensional space.

In the final step, it is necessary to compute the threshold region for the scores. In this section, it is assumed that only the first two principal components ($a = 2$) are retained in the PCA model. For this case and assuming that the data obeys a normal distribution, the confidence region, for the scores, is limited by an ellipse (threshold) according to the *T2* statistics (Chiang, et al., 2001), and is given by

$$\frac{t_1^2}{\lambda_1} + \frac{t_2^2}{\lambda_2} \le \mathrm{T}_\alpha^2 \qquad \text{Eq. 3.30}$$

$$\mathrm{T}_\alpha^2 = \frac{a\,(n\text{-}1)\,(n\text{+}1)}{n\,(n\text{-}a)}\,\mathrm{F}_\alpha(a,\, n\text{-}a) \,.$$

In (Eq. 3.30), $\{t_1; t_2\}$ are the projections along the orthogonal axes defined by the two main loading vectors, and $\{\lambda_1; \lambda_2\}$ are the eigenvalues of greater variance. The *T2* statistics threshold is T_α^2, and $\mathrm{F}_\alpha(a, n\text{-}a)$ can be obtained from a Fisher's *F*-distribution table for a certain level of significance α (0.05, or other) (Jackson, 2003; Montgomery, 1991). This PCA reference model for the data captured in nominal operation (fault-free) must be saved, in order to be used later for on-line fault detection.

The mean value and the standard deviation of the residuals computed based on the *Q* statistics must also be obtained off-line, and saved to be used later for on-line fault detection. The square of prediction error (SPE) $q(k) \in \Re^{1\times 1}$ associated with the *Q* statistics, computed at each time instant k, is given by

$$q(k) = \mathbf{e}(k)\,\mathbf{e}(k)^{\mathrm{T}} \qquad \text{Eq. 3.31}$$

where $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$, and $\mathbf{x}(k)$ is a row vector given by $\mathbf{x}(k) = \mathbf{X}(k,:) \in \Re^{1 \times m}$.

Next, the new methodology proposed here for fault detection in on-line operation is presented. The fault detection approach compares the on-line scores with the nominal elliptical region computed off-line based on the PCA reference model. A fault causes a change in the ARX model parameters and consequently a change on the correlation matrix of the PCA model.

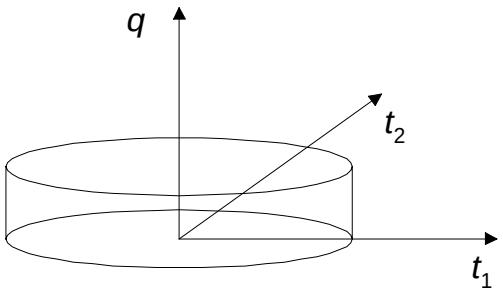


Fig. 3.11 - Fault detection using the *T2* and *Q* statistics.

The scores subspace (here, a two dimensional space) and the SPE associated with the residual subspace (one dimensional space) of the PCA model are used to define a three dimensional space, as can be observed in Fig. 3.11. In on-line operation, a fault alarm is generated if the point $P(k) = (t_1(k), t_2(k), q(k))$, with coordinates given by the scores and the square of prediction error, fall outside the ellipsoidal nominal (fault-free) region. The fault detection signal is obtained from the fault alarm signal via low pass filtering.

Next, the algorithm for fault detection that summarizes the main ideas is presented. A reference PCA model is constructed for nominal operation, i.e., for fault F_0.

Algorithm 4. Fault detection based on PCA applied to ARX model parameters (FDE-PCA-ARX).

In off-line operation, the following tasks must be executed:

a. Capture informative data (ARX model parameters, $\theta(k)$) from nominal operation, and standardize the data in order to obtain data with zero mean and unity variance. Save the mean μ and the variance σ of this nominal data for further use on-line.

b. Build a nominal PCA model M_{PCA} for the standardized data, assuming a two dimensional scores space, $a = 2$. Save this model for further application on-line.

c. Determine the thresholds and the low pass filters parameters, in order to obtain a desired trade-off between rate of false alarms, rate of missed fault detections, and detection and isolation delays. Different sets of experimental nominal data must be used to compute and validate the thresholds and filters parameters.

Each time instant k, the following steps must be executed on-line:

1. Sample the process output signal $y(k)$.
2. Estimate the ARX model parameters, vector $\boldsymbol{\theta}(k)$, using the sliding window SW-PCR parameter estimation algorithm.
3. Define the input data vector, given by the ARX model parameters $\mathbf{x}(k) = \boldsymbol{\theta}(k)$, for the PCA model M_{PCA}.
4. Standardize (via auto-scaling) the data vector $\mathbf{x}(k)$, using the mean μ and the standard deviation σ captured off-line from the nominal data, obtaining the auto-scaled data $\mathbf{x}_d(k) \in \Re^{1 \times m}$.
5. Compute the scores $\mathbf{t}_a(k) \in \Re^{1 \times a}$ using the PCA model M_{PCA}. Assuming that $\mathbf{P}_a \in \Re^{m \times a}$, for the nominal PCA model M_{PCA}, is a matrix containing the columns of the loading matrix corresponding to the loading vectors associated with the first a eigenvalues and $\mathbf{t}_a(k) = [t_1(k)\ t_2(k)]$, the scores are given by

$$\mathbf{t}_a(k) = \mathbf{x}_d(k)\ \mathbf{P}_a \qquad \text{Eq. 3.32}$$

6. Estimate the data $\mathbf{x}_d(k) \in \Re^{1 \times m}$ computing the signal $\hat{\mathbf{x}}_d(k) \in \Re^{1 \times m}$, using the nominal PCA model M_{PCA}:

$$\hat{\mathbf{x}}_d(k) = \mathbf{t}_a(k)\ \mathbf{P}_a^{\mathrm{T}} \qquad \text{Eq. 3.33}$$

7. Transform the data $\hat{\mathbf{x}}_d(k)$ to non auto-scaled obtaining the data $\mathbf{x}_e(k)$.

8. Compute the residual (prediction error), $\mathbf{e}(k) \in \Re^{1 \times m}$, by $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{x}_e(k)$, and the square of prediction error (SPE), $q(k) \in \Re^{1 \times 1}$, given by:

$$q(k) = \mathbf{e}(k)\ \mathbf{e}(k)^{\mathrm{T}} \qquad \text{Eq. 3.34}$$

9. Generate a fault alarm signal when a violation of the threshold (two dimensional elliptical region) for the *T2* statistics occurs, i.e., $am_2(k) = 1$ if

$$\frac{t_1^2(k)}{\lambda_1} + \frac{t_2^2(k)}{\lambda_2} > T_\alpha^2 \qquad \text{Eq. 3.35}$$

10. Generate a fault alarm signal when a violation of the threshold for the *Q* statistics (SPE) occurs, i.e., if $q(k) > h_q$ then $am_q(k) = 1$ else $am_q(k) = 0$.
11. A fault alarm signal, $a_m(k)$, is computed by weighted sum of the two alarms. Assuming equal weights, the fault alarm signal is expressed by $a_m(k) = [am_2(k)\ am_q(k)] \times [0.5\ 0.5]^T$.
12. Compute the fault detection signal by low pass filtering, $H_{lp}(z, \lambda)$, the fault alarm signal, and by thresholding. The thresholding is expressed by the rule: if $a_m(k) > h_f$ then $f_d(k) = 1$ else $f_d(k) = 0$. A typical value for the threshold is 0.49, i.e., a fault is detected if at least one alarm, $am_2(k)$ or $am_q(k)$, is activated during a certain time interval.

■

In the PCA approach, the dimension of the reduced space *a* is equal to the dimension of the scores space. One way to define this dimension is to choose a dimension that explains a certain percentage of the total variance of the features data (parameters), for example 80 % or more. For many applications, only two or three principal components are retained in the PCA model (Chiang, et al., 2001).

For a PCA model retaining only two principal components ($a = 2$), the next example shows a typical relation that exists between the explained variance ($E_{\sigma 2}$) and the number *m* of data variables, in this case the ARX parameters: an increase in *m* corresponds to a decrease of $E_{\sigma 2}$. The amount of variance explained by *a* principal components for a PCA model is given by (Chiang, et al., 2001; Jackson, 2003):

$$E_{\sigma 2}(a) = \sum_{i=1}^{a} \frac{\mathbf{S}_d(i, i)}{\Gamma(\mathbf{S}_d)} . \qquad \text{Eq. 3.36}$$

In Eq. 3.36, $\mathbf{S}_d \in \Re^{m \times m}$ is a diagonal matrix with nonnegative diagonal elements (eigenvalues) in decreasing order obtained from singular value decomposition of the correlation matrix $\mathbf{S} \in \Re^{m \times m}$. The function $\Gamma(.)$ represents the trace of the matrix.

Example 4. Explained variance by a PCA model with two principal components, $a = 2$.

In this example, the PCA analysis is applied to the parameters of ARX models of different orders, for data captured in nominal operation. The main goal is to compute the amount of variance explained by a PCA model that retains only two principal components ($a = 2$).

For a plant expressed by a first order system with transfer function given by $G_0(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1}$, with parameters $K = 1$ and $\tau = 1$ s, a set of ARX models with different orders are used here for discrete modeling. The ARX model parameters are obtained by closed-loop identification via the SW-PCR algorithm, with a sliding window length of 10 s. A PI controller, with $K_p = 0.4$ and $T_i = 0.4$ s, is used to control the plant. The variance of the sensor noise is 1×10^{-8}, and the reference dither signal variance is 1×10^{-3}. The sampling period is $T_s = 0.11$ s.

An ARX(n_a=1, n_b=1, n_d=1) model, with parameters $[a_1\ b_1] = [-8.9\times10^{-1}\ 1.0\times10^{-1}]$, is obtained if the zero-order hold method is used for the discretization of the transfer function $G_0(s)$. In this example, ARX(n_a, n_b=1, n_d=1) models are used for modelling, and the PCA approach is applied to capture the variability in the ARX parameters. The number of parameters is given by $m = n_a + n_b$, and two principal components ($a = 2$) are retained by the PCA model. The duration of the experiment is 400 s. The stationary data used for this example is from 10% to 100% of the experiment, in order to avoid the initial transient behaviour of the ARX parameters. The results obtained are summarized in Tab. 3.2.

Tab. 3.2 - PCA explained variance for a first order plant.

ARX(n_a, n_b=1, n_d=1) model	m	E_{σ^2} [%] with $a = 2$
ARX(1, 1, 1)	2	$\cong 100$
ARX(2, 1, 1)	3	99.82
ARX(3, 1, 1)	4	96.68
ARX(4, 1, 1)	5	85.47

Comments about the Example 4. This example shows that for a given plant an increase in the number of the ARX model parameters, due to a high order model, causes a decrease in the amount of variance explained by the PCA model with only two principal components ($a = 2$).

The performance of the proposed fault detection approach depends on the amount of variance explained ($E_{\sigma 2}$) by the PCA model. After a decision about a minimum percentage of $E_{\sigma 2}$ acceptable, for example 80 %, an appropriate ARX(n_a, n_b, n_d) model must be selected for fault detection. An appropriate ARX model for fault detection is typically a model with an order smaller than the true model of the system, and one with small variances on the parameters.

Instead of using all ARX parameters $\boldsymbol{\theta} \in \Re^m$ for fault detection, it is possible to use only a subset of parameters $\boldsymbol{\theta}_p \subset \boldsymbol{\theta}$, with $\boldsymbol{\theta}_p \in \Re^p$ and $p < m$. A method proposed here to choose the set $\boldsymbol{\theta}_p$ is based on the influence matrix (IMX, described in section 3.4.2). Since the influence matrix $\boldsymbol{\Omega} = \boldsymbol{\Omega}_{\theta,\gamma}$ is the Jacobian of the model parameter vector with respect to physical parameters, the set $\boldsymbol{\theta}_p$ must include the parameters most sensitive to the faults, i.e., the ones with high slopes in the influence matrix.

3.6 Fault Detection and Diagnosis based on a Combined Approach (PCA & IMX)

3.6.1 Introduction

Since, in most cases, only one type of FDD approach is not sufficient to detect and diagnose correctly one set of faults, the hybrid techniques combining different approaches are usually needed and used, in order to guarantee a reasonable performance and robustness (Isermann, 1997 & 2004; Palma, et al., 2005d).

In this work, for the case of linear systems, a new combined approach for fault detection and diagnosis was proposed (Palma, 2005c). One FDD approach is used for fault detection, and another for fault diagnosis.

3.6.2 The Combined FDD-PCA-IMX Approach

A new combined fault detection and diagnosis approach (FDD-PCA-IMX) for linear systems is proposed in this work based on previous works (Palma, 2005c). The overall architecture is depicted in Fig. 3.12.

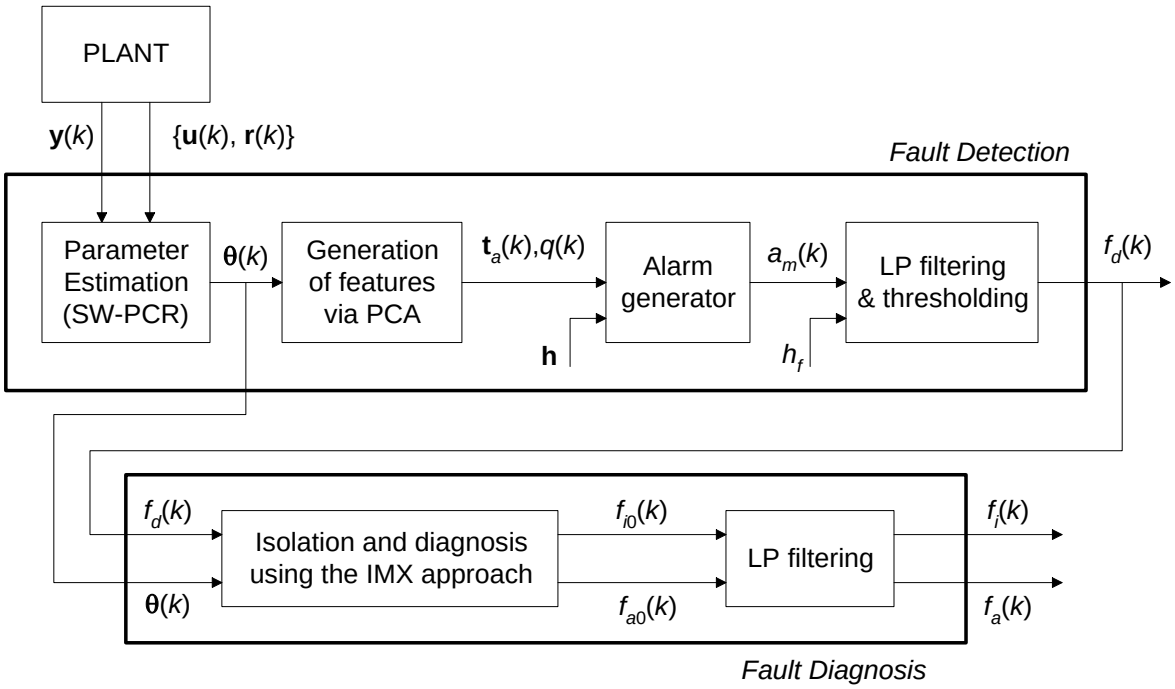


Fig. 3.12 - Architecture of the FDD combined approach (FDD-PCA-IMX).

This FDD approach was developed to be applied on-line, in real-time environments. Three main tasks can be observed: a) the on-line parameter estimation task; b) the fault detection (FDE) task; c) the fault diagnosis (FDG) task. This FDD approach is suitable for detection and diagnosis of parametric (multiplicative) process faults, but can also be used for additive faults on sensors and on actuators.

The FDD-PCA-IMX approach combines the proposed fault detection approach based on PCA analysis applied to the ARX parameters (described in section 3.5.2), with the diagnosis approach based on the influence matrix (described in section 3.4.2). First, the fault is detected, and after is isolated and its fault magnitude is estimated.

The algorithm of the proposed FDD-PCA-IMX approach is based on the architecture depicted in Fig. 3.12 and is presented next. It is a combination of Algorithm 4 for fault detection, and Algorithm 3 for fault diagnosis, both previously explained. A simplified version is presented here, in order not to repeat the various steps.

Algorithm 5. Fault detection and diagnosis combined approach based on PCA & IMX.

In off-line operation, the following tasks must be executed:

a. Build a PCA model M_{PCA} based on data captured from nominal operation, i.e., for fault F_0.

b. For the faults of the set $F_f = \{F_1, \ldots, F_n\}$ determine the influence matrix IMX, $\boldsymbol{\Omega}$.

Each time instant k, the following steps must be executed on-line:

1. Sample the process output signal $y(k)$.
2. Estimate the vector of ARX model parameters, $\boldsymbol{\theta}(k)$, using the sliding window SW-PCR algorithm.
3. Based on the nominal PCA model, M_{PCA}, compute the features for fault detection, i.e., the scores $\mathbf{t}_a(k) = [t_1(k)\ t_2(k)]$, and the square of prediction error (SPE) $q(k)$.
4. Generate a fault alarm $a_m(k)$ by comparison of the scores and SPE with the respective thresholds. The vector of thresholds is termed here **h**.
5. Compute the fault detection signal $f_d(k)$ by low pass filtering via $H_{lp}(z, \lambda)$ the fault alarm signal $a_m(k)$, and by thresholding. The thresholding is expressed by the rule: if $a_m(k) > h_f$ then $f_d(k) = 1$ else $f_d(k) = 0$.
6. If a fault is detected, i.e. if $f_d(k) = 1$, then isolate the fault according to the minimum angle criteria formulated in Theorem 3.
7. If a fault is detected, i.e. if $f_d(k) = 1$, then estimate the fault magnitude according to Eq. 3.18.

■

3.6.3 Example for the Combined Approach (PCA & IMX)

Below, an example of application of the proposed combined approach (PCA & IMX) for fault detection and diagnosis is given.

Example 5. A combined fault detection and diagnosis approach based on PCA & IMX applied to a first-order system.

The purpose of this example is to show the application of the proposed combined fault detection and diagnosis approach based on PCA & IMX. The fault detection approach is based on Algorithm 4, and the fault diagnosis is based on Algorithm 3.

In this example, it is assumed that a plant is expressed by a first order model in continuous time with the transfer function $G_0(s) = Y(s) / U(s) = K / (\tau s + 1)$. For the nominal operation the plant parameters are $K = 1$, $\tau = 1$ s. An ARX(n_a=1, n_b=1, n_d=1) model, with parameters $[a_1\ b_1] = [-8.96\times10^{-1}\ 1.04\times10^{-1}]$, is obtained if the zero-order hold method is used for the discretization of the transfer function $G_0(s)$, for a sampling period of $T_s = 0.11$ s. Two ARX models are used for discrete modeling. An input-output ARX(2, 1, 1) model, relating the output signal $y(k)$ and input signal $u(k)$, is used for fault detection via PCA applied to model parameters; a higher order is used here, since the goal is a dimensional reduction via PCA

from three to two dimensions (2 principal components). For fault diagnosis based on the influence matrix (IMX) method, an ARX(1, 1, 1) model is more appropriate, since the parameters present a smaller variance. Both ARX models, relating the input signal $u(k)$ and the output signal $y(k)$, are identified on-line using a sliding window PCR algorithm, in closed-loop operation. A digital controller is used based on the PI algorithm, with gains $K_p = 0.4$ and $T_i = 0.4$ s obtained via pole-placement, (Astrom & Hagglund, 1988).

First, for the nominal operation, PCA and IMX models are constructed in off-line operation. All the signals, set point and the input-output signals, are scaled to the range $[0; 1]$. The operating conditions are: a) reference signal (set point) is 0.5; b) the duration of experiment is 400 s; c) dither on reference signal with variance 1×10^{-3}; d) the sensor noise has a variance of 1×10^{-8}.

The nominal PCA model presents the following characteristics, assuming $a = 2$ principal components. It was created based on parameter data $[a_1\ a_2\ b_1]$ of ARX(2, 1, 1) model. PCA allows a reduction from 3 to 2 dimensions (scores space). The data has been standardized (auto-scaled) in order to have zero mean and unit variance. Each principal component explains the following variance, $[8.7879\times10^{-1}\ \ 1.1940\times10^{-1}\ \ 1.8195\times10^{-3}]$; this justifies the reduction to two dimension, since the two principal components explain more than 99 %. The threshold used for the *T2* statistics is given by $T_\alpha^2 = \frac{a\,(n\text{-}1)\,(n+1)}{n\,(n\text{-}a)} F_\alpha(a, n\text{-}a)$, according to Eq. 2.9 assuming that data obeys a normal distribution. Here, the value T_α^2 has been multiplied by a factor K_f in order to decrease the rate of false alarms, since the data does not obey a perfectly normal distribution. The value of T_α^2 used is 12.2, for $a = 2$, $n > 120$, $\alpha = 0.05$, $F_\alpha(a, n\text{-}a) = 3$, and $K_f = 2$. The threshold for the SPE has been computed based on a three sigma limit (control chart) approach given by $[\mu - 3\sigma;\ \mu + 3\sigma]$, where μ is the mean value and σ the is the standard deviation, since the threshold for the Q statistics proposed by Jackson & Mudholkar (1979) does not give satisfactory results.

The IMX model was constructed varying each physical parameters (K, τ) in the range [0.6 0.8 1.0 1.5 2.0], maintaining the other fixed at nominal value. The influence matrix (Jacobian) obtained is given by

$$\boldsymbol{\Omega} = \left[\frac{\partial \boldsymbol{\theta}}{\partial \gamma_1} \cdots \frac{\partial \boldsymbol{\theta}}{\partial \gamma_p}\right]_{\gamma=\gamma^{\text{nom}}} = \begin{bmatrix} \frac{\partial a_1}{\partial K} & \frac{\partial a_1}{\partial \tau} \\ \frac{\partial b_1}{\partial K} & \frac{\partial b_1}{\partial \tau} \end{bmatrix} = \begin{bmatrix} 5.0\times10^{-5} & -7.58\times10^{-2} \\ 1.04\times10^{-1} & -7.58\times10^{-2} \end{bmatrix}. \qquad \text{Eq. 3.37}$$

The relationship between ARX model parameters and physical parameters can be observed in Fig. 3.13. The elements of the influence matrix are the slopes of the linear regression of the data. Theoretically, in the IMX method it is assumed that "the model parameters are multi-linear in the physical parameters". In practice this is not perfectly true (see Fig. 3.13).

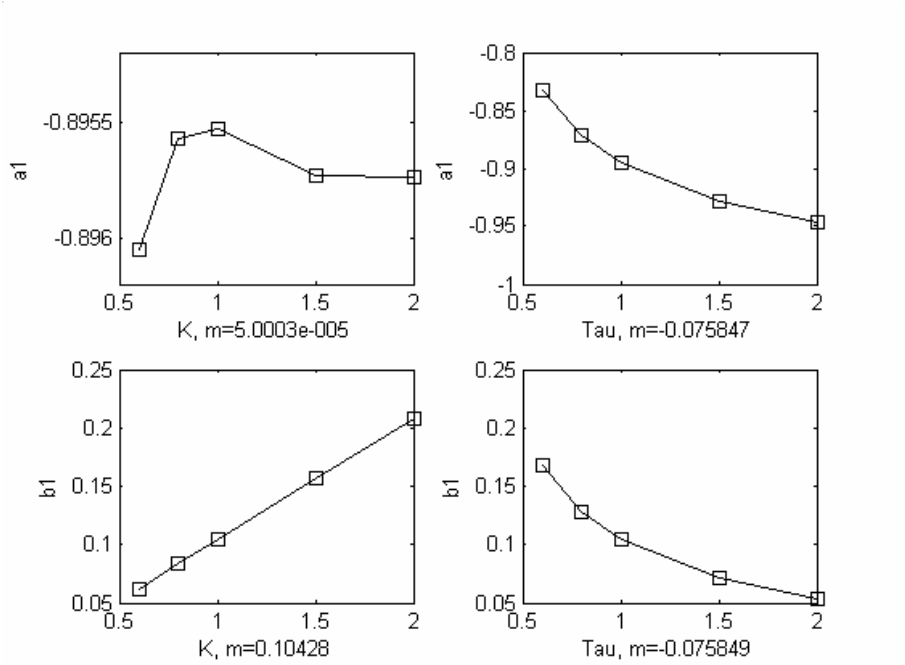


Fig. 3.13 - ARX model parameters versus physical parameters.

Afterwards two experiments carried out using the combined PCA/IMX method for fault detection and diagnosis are shown. Two faults are considered: a) fault F_1, a change on the gain K from 1 to 2; b) fault F_2, a change on the time constant τ from 1 to 2.

Next, the experiment carried out with the fault F_1 is presented, a change on the gain K from 1 to 2. In Fig. 3.14 can be observed from top to bottom: the reference signal $r(k)$, the output signal $y(k)$, the control input $u(k)$, the alarm signal $a_m(k)$ and other signals (see the explanation next), the fault detection signal $f_d(k)$, the fault isolation signal $f_i(k)$, and finally the fault analysis signal $f_a(k)$. In the graph with label "am" it can be observed, from top to bottom, the alarm signal $a_m(k)$, the fault trigger signal, the *T2* statistics signal, and the *Q* statistics signal. In the "fd" graph, it can be observed the fault detection signal $f_d(k)$, and the non-filtered signal $f_{d0}(k)$ in dotted line.

The fault occurs at time instant 240 s. The fault detection delay is 0.7 s, and the correct fault isolation delay is approximately 12 s (computed from the time of fault occurrence). A wrong isolation occurs after fault detection, due to the transient of the ARX model parameters. After

that, the fault is well isolated, and the estimation of the fault magnitude is correct (the value is near 1, see $f_a(k)$ signal) since the relation between the model parameters (a_1, b_1) and the physical parameter (K) is approximately linear, as can be observed in Fig. 3.13.

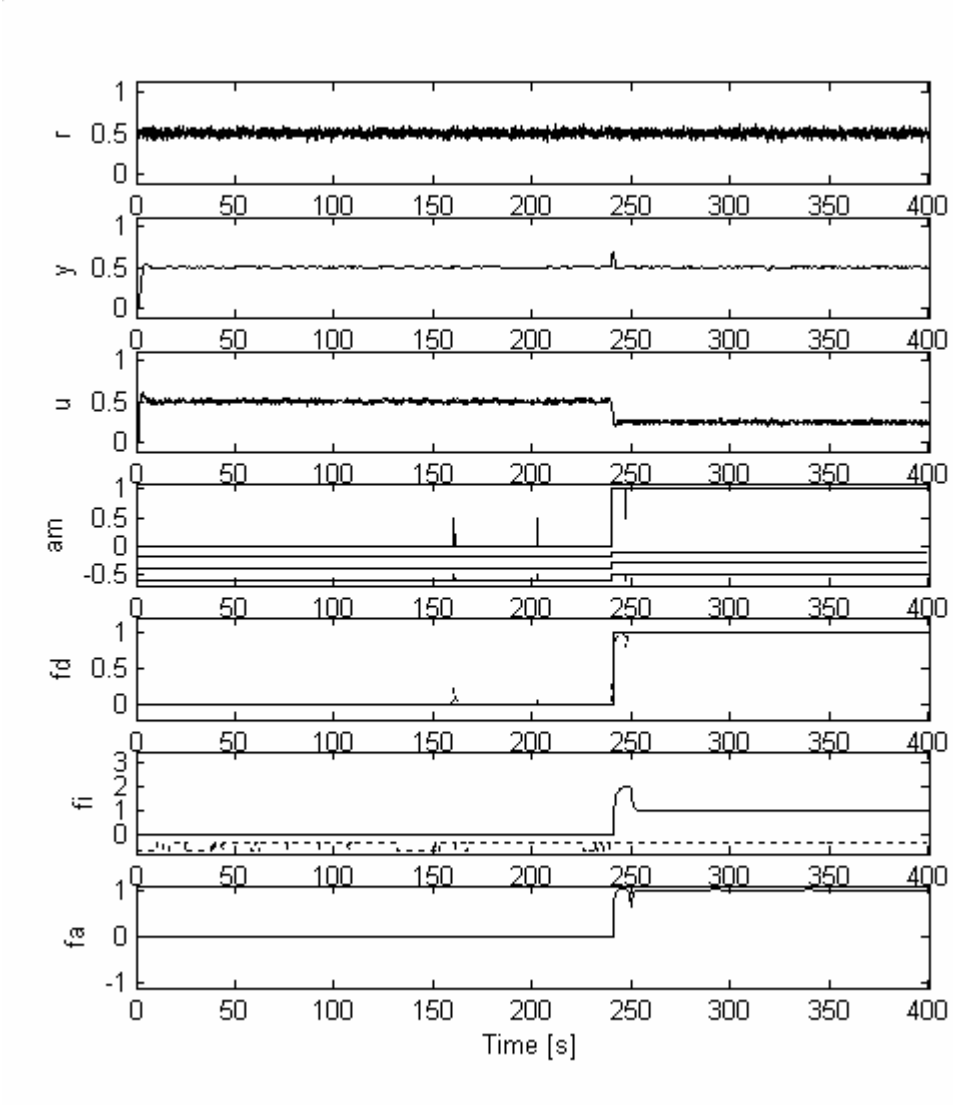


Fig. 3.14 - Input-output and FDD signals for fault F1.

Fig. 3.15 shows the evolution of the physical parameters (K, τ), and the evolution of the parameters for both ARX models. The parameters of ARX(2, 1, 1) model are depicted in the graph "th211"; the nominal values are given by $[a_1\ a_2\ b_1] = [-8.90\times10^{-1}\ -5.22\times10^{-3}\ 1.04\times10^{-1}]$. The graph "th111" shows the parameters of ARX(1, 1, 1) model; the nominal values are given by $[a_1\ b_1] = [-8.96\times10^{-1}\ 1.04\times10^{-1}]$.

The wrong isolation during the first instants after fault detection is well understood here, observing the evolution of the model parameters in the graph "th111".

The transient time on the ARX model parameters after a fault occurrence is equal to the length of the sliding window of the SW-PCR algorithm, i.e., a value of 10 s.

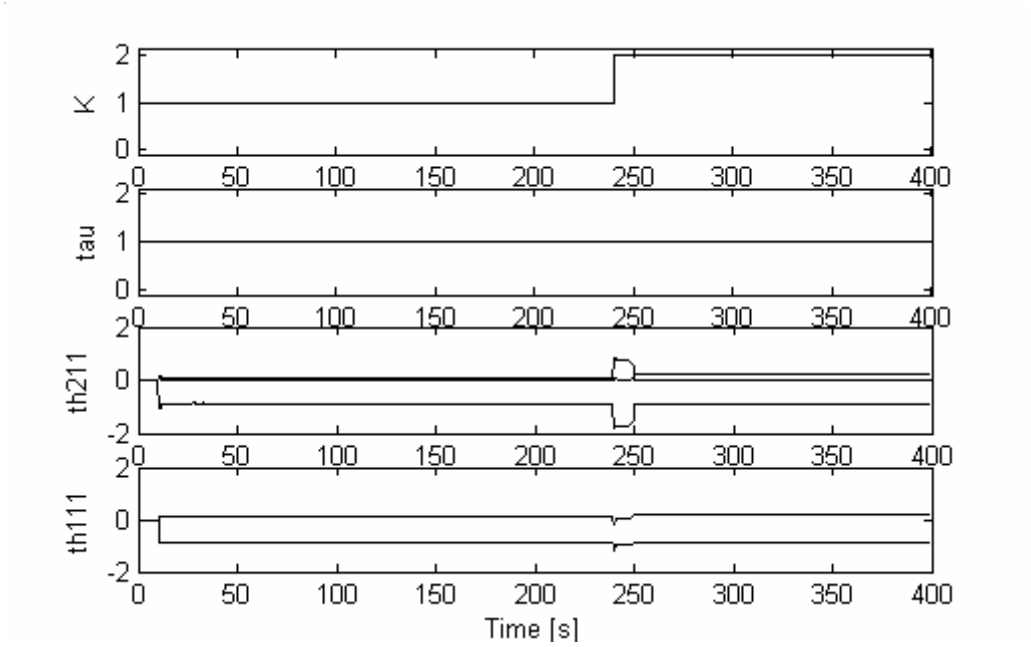


Fig. 3.15 - Model parameters transient for fault F1.

An experiment carried out with fault F_2, a change on the time constant τ from 1 to 2, is now presented. The fault detection and isolation performance, for this case, is better than for the case of fault F_1, because the effect of this fault is softer on the transient behaviour of the ARX model parameters, θ.

In Fig. 3.16 it can be observed, from top to bottom: the reference signal $r(k)$, the output signal $y(k)$, the control input $u(k)$, the alarm signal $a_m(k)$ and other signals (see the explanation next). Next, the fault detection signal $f_d(k)$, the fault isolation signal $f_i(k)$, and finally the fault analysis signal $f_a(k)$ appears. In the graph of the alarm signal $a_m(k)$, can be observed from top to bottom, the alarm signal $a_m(k)$, the fault trigger signal, and signals that indicate a violation of the *T2* statistics and the *Q* statistics. In the "fd" graph, it can be observed the signal $f_d(k)$, and the non-filtered signal $f_{d0}(k)$ in dotted line. The fault detection delay is approximately 2 s, and the correct fault isolation delay is approximately 5 s. The fault is well isolated after fault detection since the parameters change in the correct direction. The estimation of the fault magnitude is not correct (the value is approximately 0.7, less than the correct value of 1.0, see $f_a(k)$ signal) since the relation between the model parameters (a_1, b_1) and the physical parameter τ are not linear enough, as can be observed in Fig. 3.13.

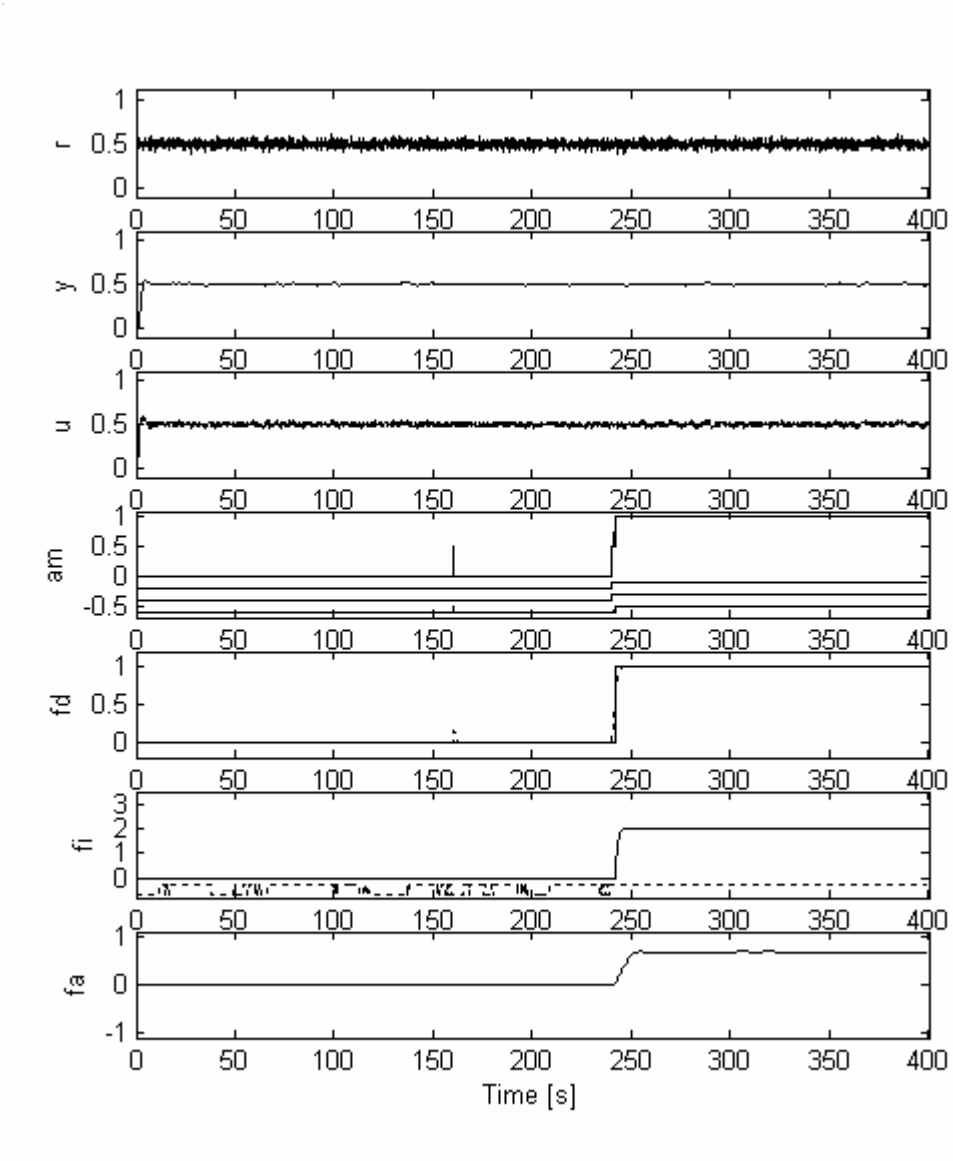


Fig. 3.16 - Input-output and FDD signals for fault F2.

In Fig. 3.17 it can be observed, from top to bottom, signals obtained via the PCA model and used for fault detection: the scores space ($t_1(k)$, $t_2(k)$) monitored by the *T2* statistics, the first principal component $t_1(k)$, the second principal component $t_2(k)$, and the square of prediction error $q(k)$ (*Q* statistics). When a fault occurs, in this case both statistics are violated.

In Fig. 3.18, the IMX angles used for fault isolation can be observed. The signal *ang01*(k) is the angle between $\Delta\boldsymbol{\theta}(k)$ and the first influence vector $\frac{\partial \boldsymbol{\theta}}{\partial K}$. The signal *ang02*($k$) is the angle between $\Delta\boldsymbol{\theta}(k)$ and the second influence vector $\frac{\partial \boldsymbol{\theta}}{\partial \tau}$. Here, the fault F_2 is well isolated according to the minimum angle criterion (Theorem 3).

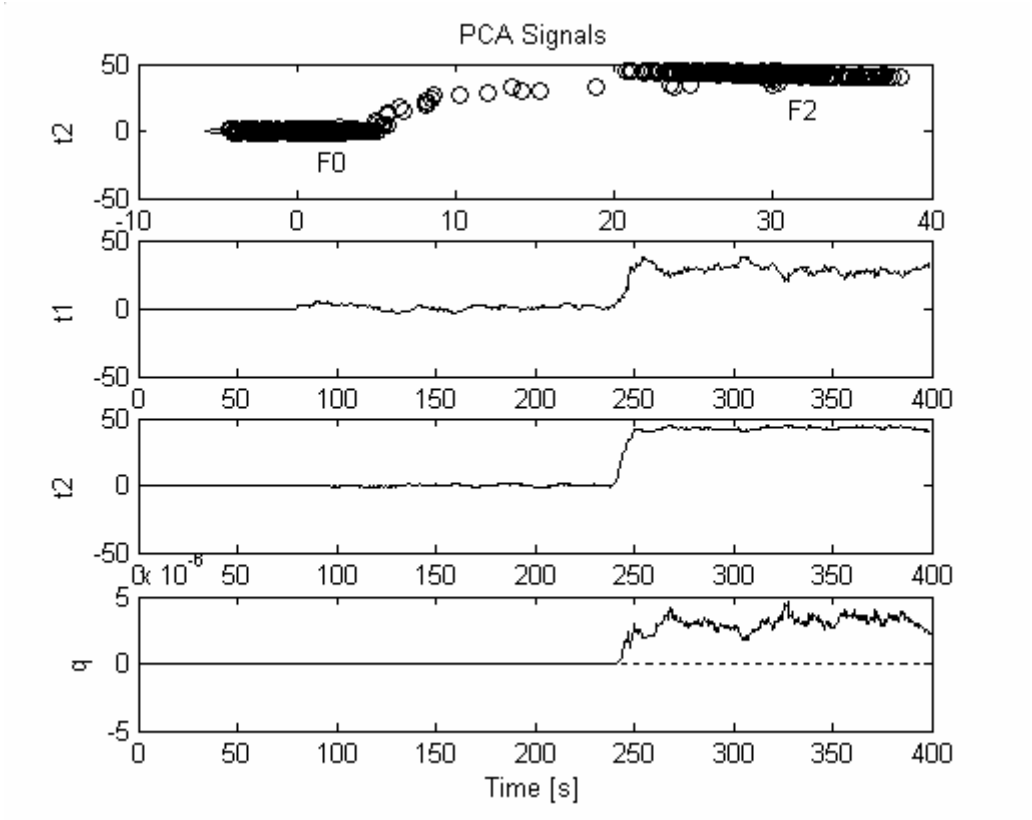


Fig. 3.17 - PCA signals for fault F2.

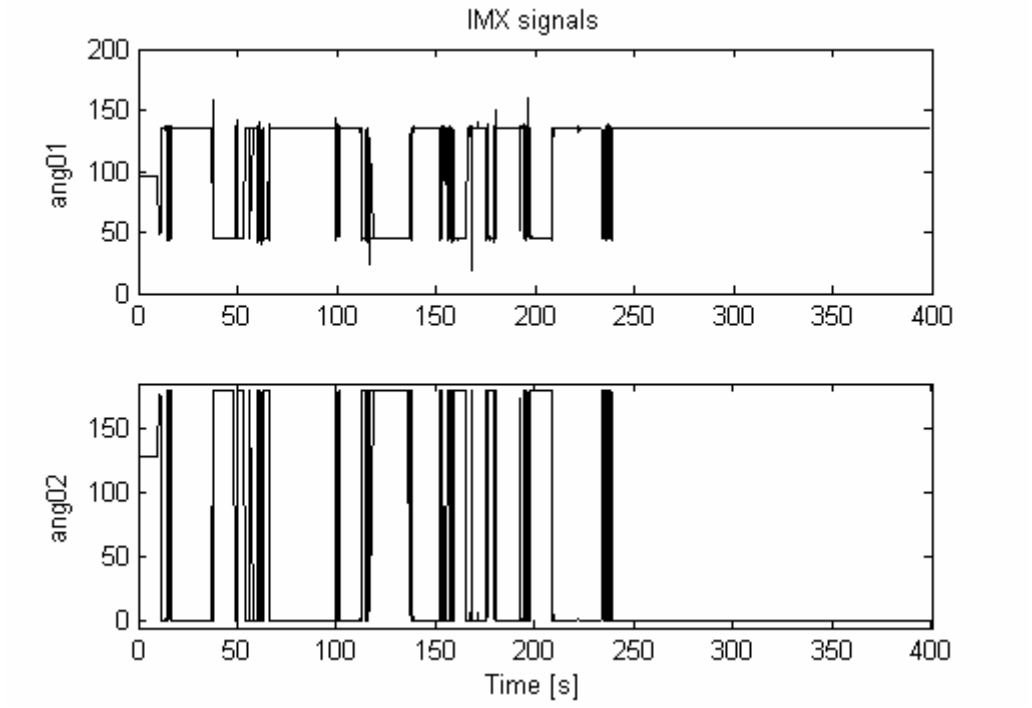


Fig. 3.18 - IMX signals for fault F2.

Finally, Fig. 3.19 shows the temporal behaviours of the ARX model parameters. It is clear that this fault causes a smooth transient in the model parameters. From top to bottom, the physical parameters (K, τ), and the parameters of both models, ARX(2, 1, 1) model and ARX(1, 1, 1) model, are depicted.

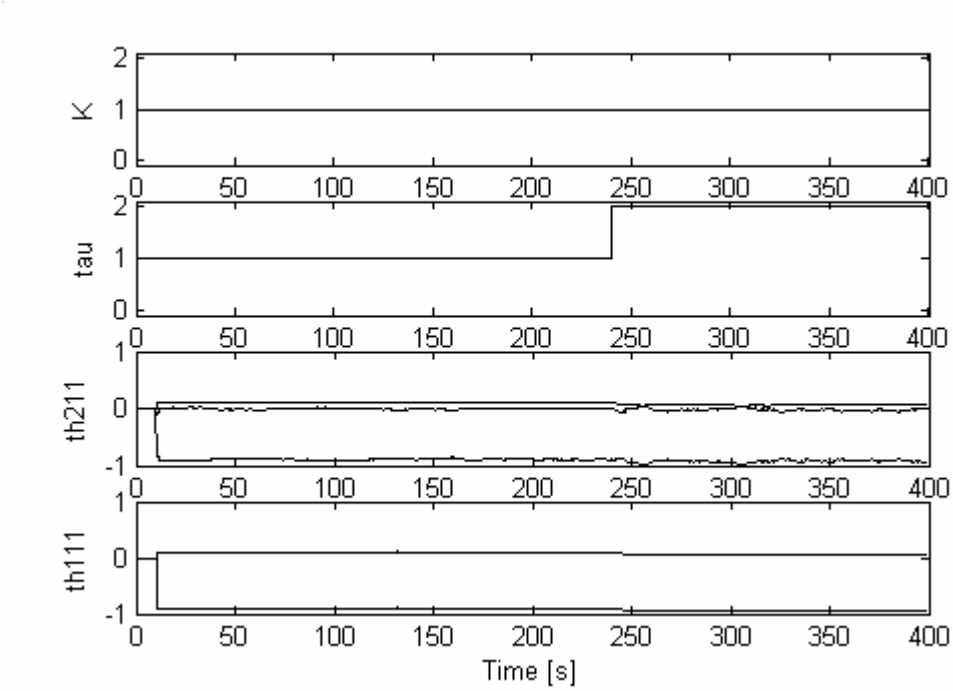


Fig. 3.19 - Parameters transient for fault F2.

3.6.4 Dimensionality Analysis of Features Space

In this section, a dimensionality analysis of the features space for both the PCA approach for fault detection, and for the IMX approach for fault diagnosis is made.

The scores contain information related to the differences between the objects (variables, features, etc). If the goal is to classify objects then uni-dimensional or bi-dimensional scores graphs are the most appropriate (Lopes, 2001).

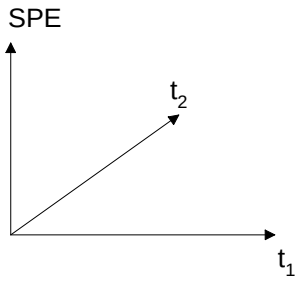


Fig. 3.20 - Three dimensional plot for scores and SPE.

For the PCA approach for fault detection, a two dimensional scores space based on the *T2* statistics is usually sufficient, if the *Q* statistics (also kwnon as the square of prediction error, SPE) is also used. Using both statistics a three dimensional features space for fault detection is obtained, as shown in Fig. 3.20.

A theoretical rule for the selection of the number of principal components in a PCA model does not exist, but several empirical rules exist (Chiang, et al., 2001; Lopes, 2001; Jolliffe, 2002; Jackson, 2003). The most common empirical rules are: the percent variance test, the PRESS statistics, the correlation analysis, the cross validation. In this work, the percent variance test is used (Chiang, et al., 2001; Jackson, 2003); this method determines a by calculating the smallest number of loading vectors ("principal components") needed to explain a specific minimum percentage of the total variance (generally, greater than 80 %). The first a eigenvalues, $\lambda_1 \geq \ldots \geq \lambda_a$, explains the greatest variance, and the corresponding loading vectors define the scores space.

For the IMX approach for fault diagnosis (isolation and analysis) a method to determine the model parameters to use in the influence matrix, $\mathbf{\Omega}_{\theta,\gamma}$, is proposed here based on the following rules:

a) The sensitivity to faults, $\partial\boldsymbol{\theta}_j/\partial\gamma_i$, must be as high as possible;

b) The influence vectors, $\mathbf{\Omega}_i$, must be well separated in the Euclidean space of feature vectors;

c) The features must be defined in a two or three dimensional space, in order to facilitate geometrical visualization, validation and interpretation. A three dimensional graph must be decomposed in three two dimension graphs to facilitate the visualization and interpretation. The number of graphs in two dimension for a n-dimensional space can be computed according to the combinatorial formula, $C_p^n = n! / (p! \, (n\text{-}p)!)$, with $p = 2$.

3.6.5 Influence of Sensor Noise

Here, the influence of the sensor noise on the performance of the fault detection and diagnosis (PCA & IMX) approach proposed is described. It is expected that an increase of the variance of the sensor noise will cause an increase of the variance of the model parameters, as was shown experimentally in Example 2.

Since both the fault detection and the fault diagnosis approaches described in this section depend on the on-line estimated parameters, it is expected that an increase of the variance of the sensor noise will cause a decrease of the FDD performance. Two examples are given next,

in order to show the performance degradation due to the increase of the variance of the sensor noise.

Example 6. Degradation of the fault detection performance due to the increase of variance of the sensor noise.

The next table shows the results obtained for a set of five simulations done for the first order model $G_0(s) = Y(s) / U(s) = K / (\tau s + 1)$, with process parameters $K = 1$ and $\tau = 1$ s. For fault detection, an ARX(2, 1, 1) model has been used. The operating conditions are similar to the ones described in Example 5. The experiments use the thresholds for the nominal case, corresponding to a variance of sensor noise of 1×10^{-8}. For different values of variance of sensor noise, the rate of false alarms Ψ_{fa} has been computed and the mean value is shown.

Tab. 3.3 - Rate of false alarms versus variance of sensor noise.

Variance of sensor noise	$\mu(\Psi_{fa})$ [%]
1×10^{-8}	2.9
1×10^{-7}	80.1
1×10^{-6}	100

If the variance of sensor noise increases, then to guarantee an acceptable rate of false alarms it is necessary to define a larger nominal region, and consequently to compute new threshold values. For high variances, the drawback is the impossibility of detection of small faults. The use of adaptive thresholds can be considered to deal with different sensor noise situations.

Example 7. Influence of sensor noise on the FDD-PCA-IMX diagnosis performance.

Some simulations have been performed for the first order model $G_0(s) = Y(s) / U(s) = K / (\tau s + 1)$, with process parameters $K = 1$ and $\tau = 1$ s, in order to evaluate the influence of the increase of the sensor noise on the fault diagnosis performance. For fault diagnosis, an ARX(1, 1, 1) model has been used in the IMX approach. The operating conditions are similar to the ones described in Example 5. The experiments use the thresholds for the nominal case, corresponding to a variance of sensor noise of 1×10^{-8}. The fault diagnosis task based on the IMX approach takes place after fault detection.

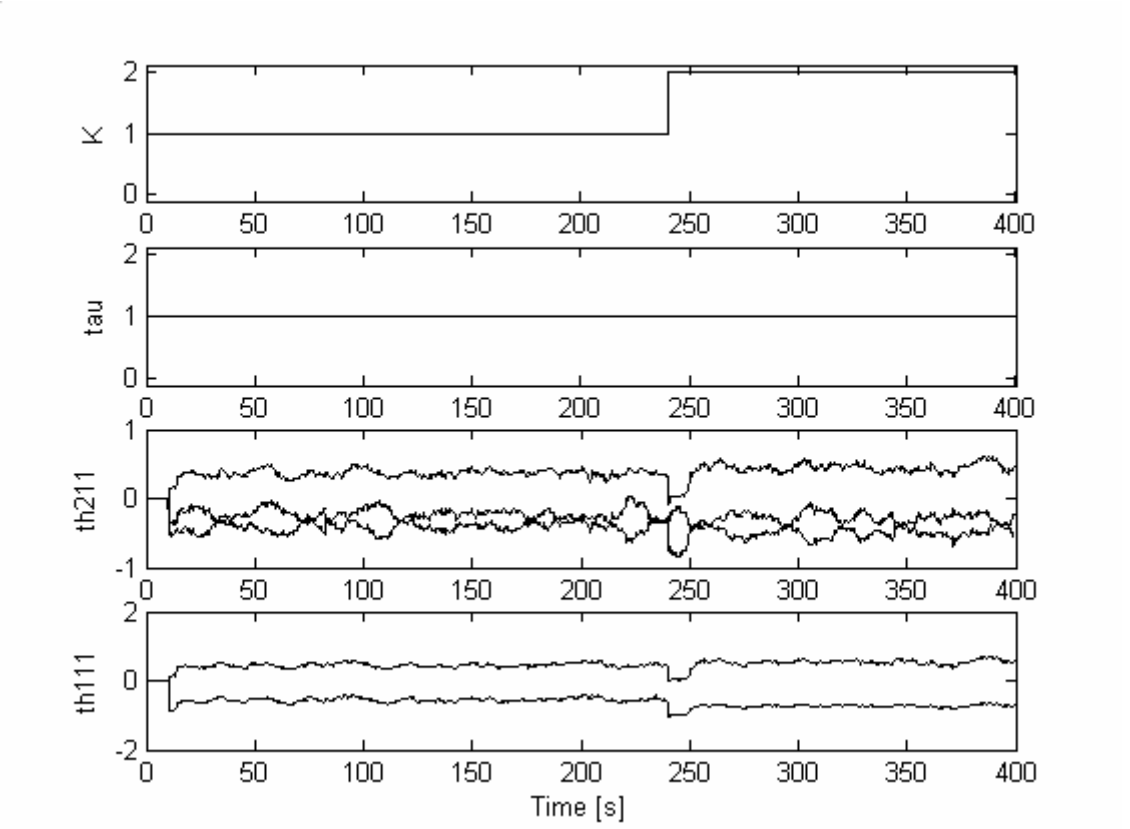


Fig. 3.21 - Parameters of ARX models for fault F1.

The fault diagnosis approach is less sensitive to the influence of noise than the fault detection approach, because the fault diagnosis uses an ARX(1, 1, 1) model while the fault detection uses an ARX(2, 1, 1) model. Model ARX(1, 1, 1) is a better model for discrete modeling of the system $G_0(s) = Y(s) / U(s) = K / (\tau s + 1)$. Fig. 3.21 shows some results obtained for a variance of sensor noise given by 1×10^{-4} (much larger than the nominal value of 1×10^{-8} previously assumed). The first two graphs show the physical parameters K and τ. In the third graph the parameters ("th211", $[a_1\ a_2\ b_1]$) of the model ARX(2, 1, 1) used for fault detection can be seen, and the last graph shows the parameters ("th111", $[a_1\ b_1]$) of model ARX(1, 1, 1) used for fault diagnosis. It is clear that the variances of the ARX(2, 1, 1) model parameters are greater than the variances of the ARX(1, 1, 1) model parameters, and this justifies the larger sensitivity to noise of the fault detection approach.

Fault isolability problems occur when noise variance is high and the angle between influence vectors is small, because noise causes variation of the angles between the influence vectors. For this situation, in certain directions, faults cannot be well isolated.

3.7 Conclusions

In this section, new fault detection and diagnosis methodologies suitable to be applied on-line in linear systems have been proposed. In most practical problems white box models are not available, or they are difficult and time consuming to obtain. Sometimes the only alternative is to use black-box models obtained via identification techniques. That is the main reason why the focus of the research done was on black box models. A good model is application-dependent and may not necessarily be the true model. The FDD approaches proposed are based on black-box ARX models and require on-line parameter estimation. The system identification is made in closed-loop, using the sliding window SW-PCR algorithm, and a small dither signal is added to the reference signal in order to guarantee good persistent excitation conditions. The proposed sliding window parameter estimation algorithm based on principal components regression shows a good performance. The FDD parameter identification methods are more suitable for detection and diagnosis of multiplicative (parametric) faults, but can also be used for FDD of additive faults on sensors and actuators.

The first new fault detection and diagnosis methodology proposed is based on system dynamic features (static gain s_g and bandwidth b_w) computed on-line from ARX models. This approach can be applied to SISO systems. Both input-output ARX models $M_{yu}(\boldsymbol{\theta})$ and reference-output ARX models $M_{yr}(\boldsymbol{\theta})$ can be used for FDD. The fault detection and isolation is based on a pattern classification approach, where the input pattern is given by the dynamic features (static gain s_g and bandwidth b_w) and the output is the fault class. A neural network implements the pattern classifier according to a typical architecture associated with a neural nonlinear discriminant analysis (NNLDA). The discriminant analysis NNLDA allows the definition of decision boundaries needed for fault detection and isolation, and is more efficient than the geometrical techniques. The experiments carried out using process models, and shown in the examples, show a good FDD performance for the class of faults tested. The performance depends strongly on the consistency of the parameter estimates. An increase in the noise level in the sensor degradates the FDD performance since there is an increase in the variance of the ARX model parameters and in the variance of the dynamic features. Hence, high levels of noise renders the detection and diagnosis of small faults more difficult.

Typically, principal components analysis (PCA) is applied to input-output data. Another new contribution given in this work is a fault detection methodology based on the application of

principal components analysis to ARX model parameters data, θ. Typically, parametric faults in the physical process parameters change the system dynamics, and their symptoms are deviations on the ARX model parameters from the nominal values. The PCA method allows the detection of these changes, and the definition of thresholds based on the *T2* statistics and the *Q* statistics. This fault detection method based on PCA has been combined with an influence matrix method for fault diagnosis, and a new combined FDD approach was proposed. This combined FDD method is appropriate for application in SISO or MIMO systems modeled by ARX models. The experiments done using process models, and shown in the examples, show a good FDD performance for the class of faults tested. Some examples related to the influence of sensor noise on the FDD performance have been presented.

The main aim of the examples presented is to elucidate the principles, and to show the advantages and drawbacks of each methodology. The FDD methodologies proposed here show a good effectiveness. The examples also show that the quality of the ARX models used is extremely important to guarantee a good FDD performance. Noise and nonlinear effects that occur in these types of FDD approaches decrease the overall performance.
For nonlinear systems, the development of new approaches to FDD is a great challenge. This subject will be discussed in the next chapter.

4 Fault Detection and Diagnosis Approaches for Nonlinear Systems

In the real world, almost all the systems are nonlinear (L. B. Palma).

4.1 Introduction

The majority of model-based fault detection and diagnosis (FDD) methods are based on linear system models. For nonlinear systems, the FDD problem has been traditionally approached in two steps (Chen & Patton, 1999): 1) the model is linearized around an operating point; 2) robust techniques are applied to generate features (residual signals, etc) which must be insensitive to model parameter variations within a small neighbourhood of the operating point. For systems with high nonlinearity and a wide dynamic operating range, the linearized approach fails to give satisfactory results. One possible solution is a multi-model approach using a large number of linearized models corresponding to a range of operating points; this is not very practical for real-time applications.

It is necessary to develop FDD methods which tackle nonlinear dynamic system models directly. Chen & Patton (1999) mentioned some references which attempt to use nonlinear classical observers to solve nonlinear system FDD problems based on adaptive observers, identity observers, unknown input observers, etc; for some cases, no methods exist yet to design the gain matrix for ensuring the observer stability. There have also been some studies on nonlinear parity equations. Unlike linear systems, there is no direct link between parity equation and observer-based FDD approaches in the case of nonlinear systems. The parameter estimation approach for FDD can also be extended to deal with nonlinear systems; a great advantage of this approach is that it captures the system dynamics on-line.

In many situations, it is very difficult to obtain the analytical models that the classical nonlinear observer FDD approaches require. Sometimes, the system cannot be modeled by explicit mathematical models. Without a model, the observer-based FDD approach cannot be

implemented. To overcome this problem, it is desirable to find a general approximate model which can be used to represent any nonlinear system approximately. The Neural Network (NN) is exactly such a powerful tool for handling nonlinear problems (Narendra & Parthasarathy, 1990; Haykin, 1994; Chen & Patton, 1999). FDD techniques based on neural networks are gaining more and more interest, mainly due to their ability to deal with nonlinear systems, and their robustness to noise (Patton, et al., 2000; Palma, et al., 2005b).

In the use of neural networks for fault detection and diagnosis (FDD) there are two major problems accompanying the majority of early publications: 1) most studies only deal with steady-state processes; 2) the NN has been used mostly as a fault classifier based on output data. To achieve on-line FDD in the presence of transient behaviours, the system dynamics have to be considered. A diagnosis method which only utilizes output information could give incorrect information about faults in the system, when the system input has been changed; this is especially true for nonlinear systems. The solution is to use the residual generation concept, in which both the input and output signals are required, combined with neural networks in order to form a powerful FDD tool for nonlinear dynamic systems. In recent years, some authors have followed this research direction, mainly using neural predictors and neural observers; some references are (Chen & Patton, 1999; Frank, et al., 2000b; Koppen-Seliger & Frank, 1995; Marcu, et al., 1998; Marcu, et al., 1999). Some references on neural observers are in fact neural predictors without recurrency. Some examples are the papers written by Koppen-Seliger et al. (1999) and by Palma et al. (2003c), and the dissertation written by Genrup (2005). There are few works related to the application of neural observers with gain matrix (Zhou & Bennett, 1997), and neural predictors with recurrency (Palma, et al., 2004a; Palma, et al., 2005b).

In some references, the term observer is used indistinctly for state observers (with gain matrix), and for output observers (in fact, the terminology used here is output predictors) with or without recurrency.

In this dissertation, the basic observer concepts are only briefly introduced since it is assumed that the analytical models are not available. Two main contributions are given here. The first is a new FDD approach based on neural recurrent output predictors, inspired by the classical observer structure, and based on nonlinear NARX neural models with external recurrency. The second is a new FDD approach based on neural nonlinear principal component analysis (neural NLPCA) applied to the parameters of ARX models, combined with a neural nonlinear discriminant analysis approach for pattern classification.

4.2 Observers for Nonlinear Systems

This section briefly reviews two main types of classical observer structures: the Luenberger observer and the Kalman filter. Intelligent neural and fuzzy observers with similar structure can also be constructed.

In practice many nonlinear systems cannot be represented by linear models, in particular when they are not working at a fixed operating point. This is the usual case in FDD, because at the occurrence of a fault the process diverges from its operating point. Therefore, a nonlinear model should be used.

For a dynamic system, $S(\mathbf{x}, \mathbf{y}, \mathbf{u})$, an observer with state $\mathbf{x}$, output $\mathbf{y}$, and input $\mathbf{u}$ is another dynamic system $\hat{S}(\mathbf{x}, \mathbf{y}, \mathbf{u})$ having the property that the state $\hat{\mathbf{x}}$ of the observer $\hat{S}$ converges to the state $\mathbf{x}$ of the process S, independent of the input $\mathbf{u}$ or the state $\mathbf{x}$ (Friedland, 1996). The concept of a classical observer for a dynamic process was introduced by Luenberger (1966). The generic "Luenberger observer", however, appeared several years after the Kalman filter (KF). In fact, the Kalman filter (Kalman, 1960; Welch & Bishop, 2001) is an important special case of a Luenberger observer – an observer optimized for the presence of noise.

For a nonlinear system, the structure of the classical observer is not so obvious as it is for a linear system (Friedland, 1996). Let us assume a nonlinear stochastic dynamic model for a nonlinear plant

$$\begin{cases} \mathbf{x}(k+1) = f_m(\mathbf{x}(k), \mathbf{u}(k), \mathbf{Q}_p(k)) \\ \mathbf{y}(k) = g_m(\mathbf{x}(k), \mathbf{u}(k), \mathbf{R}_m(k)) \end{cases} \qquad \text{Eq. 4.1}$$

where $\mathbf{x}(k) \in \Re^n$ is the state, $\mathbf{u}(k) \in \Re^u$ is the input vector, $\mathbf{y}(k) \in \Re^y$ is the output vector, and $f_m(.)$ and $g_m(.)$ are nonlinear functions. The matrices $\mathbf{Q}_p(k)$ and $\mathbf{R}_m(k)$ are the process and measurement noises. Assuming the noise characteristics are known, an Extended Kalman Filter (EKF) can be used as a nonlinear observer (Friedland, 1996; Welch & Bishop, 2001). The EKF is difficult to implement in practice, since the spectral densities matrices $\mathbf{Q}_p(k)$ and $\mathbf{R}_m(k)$ are hardly ever known to be better than an order of magnitude.

Neglecting the noise terms, $\mathbf{Q}_p(k)$ and $\mathbf{R}_m(k)$, the model (Eq. 4.1) represents a deterministic system, and for this situation the architecture of a general nonlinear Luenberger observer in

discrete-time is depicted in Fig. 4.1; this structure was adapted from the version in continuous time explained by Friedland (1996).

The observer equations are given by Eq. 4.2, and the observer gain K_n (a design parameter, fixed or adaptive) must make the state error equation, $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$, asymptotically stable:

$$\begin{cases} \hat{\mathbf{x}}(k+1) = f_m(\hat{\mathbf{x}}(k), \mathbf{u}(k)) + K_n\, \mathbf{r}_e(k) \\ \hat{\mathbf{y}}(k) = g_m(\hat{\mathbf{x}}(k), \mathbf{u}(k)) \end{cases} \qquad \text{Eq. 4.2}$$

In Eq. 4.2, $\mathbf{r}_e(k)$ is the output prediction error given by $\mathbf{r}_e(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$.

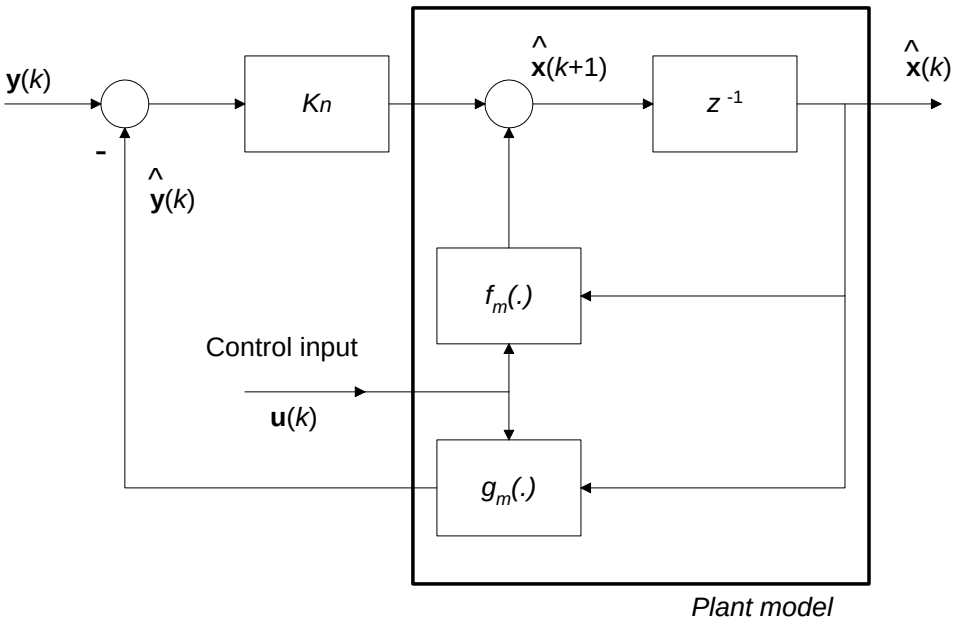


Fig. 4.1 - Structure of a discrete-time nonlinear Luenberger observer.

Since neural networks and fuzzy systems can model nonlinear systems, they can be used for implementing nonlinear state observers and nonlinear output predictors.

The next section will show how neural network models can be used for identification of nonlinear systems. Also described is how neural networks can be used for implementing neural output predictors without recurrency and with external recurrency.

The great advantages of these neural models is the great potential to model SISO or MIMO nonlinear systems, if informative data is available in the training phase.

4.3 Neural Recurrent Output Predictors (NROP)

4.3.1 Introduction

Neural networks models can be classified also as nonlinear black-box models. Artificial neural networks can be used effectively for the identification and control of nonlinear dynamic systems, and also for fault detection and diagnosis.

The identification of nonlinear systems can be made effectively by an artificial neural network, if the training data is rich enough. Two classes of neural networks (NN) which have received considerable attention in the area of artificial neural networks in recent years are the multi-layer neural networks, and the recurrent neural networks. Internal or external recurrence can be implemented in neural structures. From a systems theoretic point of view, multi-layer networks represent static nonlinear maps, while recurrent networks are represented by nonlinear dynamic feedback systems (Narendra & Parthasarathy, 1990; Hagan, et al., 1995).

Some papers were published using neural output predictors without recurrency (Koppen-Seliger, et al., 1999; Palma, et al., 2003c; Genrup, 2005).

In this section, a new neural architecture for constructing Neural Recurrent Output Predictors (NROP) based on a neural parallel model with external recurrency is presented.

4.3.2 Problem Formulation

Assuming that we are dealing with a single-input single-output (SISO) nonlinear system, without loss of generality, the general problem under study in this section can be formulated as follows.

Problem 3. Given a continuous time nonlinear SISO dynamic system modeled by a black-box nonlinear NARX neural model, find a neural structure for output prediction in real-time operation.

Typically, most of the dynamic systems in industry work under closed-loop control, as depicted in Fig. 4.2. Assuming a SISO system, the process plant is represented by the block P, and the blocks A and S represent, respectively, the actuator and the sensor. The blocks A/D

and D/A are the analog-to-digital and digital-to-analog converters. The digital computer implements the supervision, control and FDD algorithms. The digital signals are also represented in the figure: the reference signal, $r(k)$, and the input and output signals, $u(k)$ and $y(k)$, respectively. The other signals are analog signals: the real process input, $u_r(t)$, and the real process output, $y_r(t)$. The signals, $w(t)$ and $v(t)$, are, respectively, the disturbance input to the plant, and the disturbance or noise in the sensor.

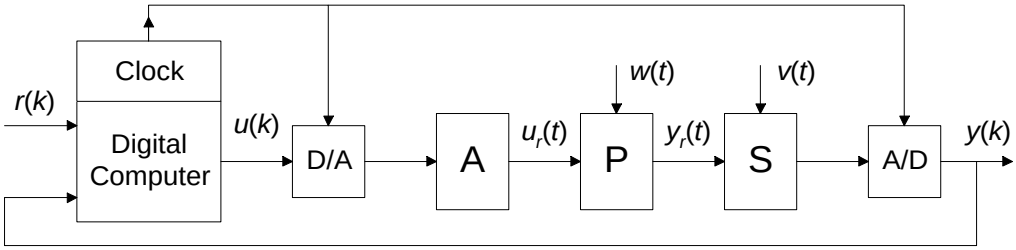


Fig. 4.2 - Closed-loop control architecture.

Let us assume that a nonlinear dynamic system is described by the SISO nonlinear NARX model (Eq. 4.3) without loss of generality, where k is the discrete-time instant, n is the system order, $u(k) \in \Re^u$ is the input vector, $y(k) \in \Re^y$ is the output vector, and $F(.)$ is a nonlinear function.

$$y(k) = F(y(k\text{-}1), \ldots, y(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) \qquad \text{Eq. 4.3}$$

Neural prediction models can be used to implement the nonlinear mapping between the input data and the output data of a dynamic process.

4.3.3 Neural Prediction Models

Neural models can be used in different types of architectures. A multi-layer perceptron feed-forward neural network (MLP-FF-NN) with weight matrix **W** can be used to model the nonlinear function expressed by the model (Eq. 4.3).

Neural networks, in the form of neural one-step ahead predictors, can be used as neural NARX output prediction models (Narendra & Parthasarathy, 1990; Palma, et al., 2003a), expressed in the form (Eq. 4.4). $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W},\ldots)$ denotes a neural network based nonlinear functional mapping, with height matrix **W**, with network structure $\{a\text{-}b\text{-}c\}$ corresponding to

the number of neurons on each layer, respectively the input-layer (IL), the hidden layer (HL), and the output layer (OL).

$$\hat{y}_{nop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, y(k\text{-}1), \ldots, y(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) \quad \text{Eq. 4.4}$$

A parallel model expressed by (Eq. 4.5) can also be developed using neural networks (Narendra & Parthasarathy, 1990), where the predicted output depends on the past predicted output values, and on the input values.

$$\hat{y}_{nopp}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nopp}(k\text{-}1), \ldots, \hat{y}_{nopp}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) \quad \text{Eq. 4.5}$$

Inspired by the neural parallel model and the observer structure with prediction and correction mechanisms, a new architecture for a neural recurrent output predictor (NROP) is proposed in this dissertation to be described below.

4.3.4 Neural Recurrent Output Predictor (NROP) with external feedback

The new neural output prediction structure was proposed by (Palma, et al., 2004a) with the aim of application in fault detection and diagnosis tasks, to deal with linear or nonlinear systems. In (Palma, et al., 2005b), the authors applied this methodology for FDD in a DC motor linear model using a bank of NROP predictors.

The main idea is to have one NROP predictor tuned to each interested operational situation (nominal operation, and faulty cases). This is achieved via nonlinear system identification, i.e., off-line training of the neural network with informative data. Each neural network embedded model is trained with informative data captured for the operational situation, and by adjusting a predictor gain.

Inspired by the neural parallel model and the observer structure with prediction and correction mechanisms, the NROP neural output predictor can be written in the form of a nonlinear function $\Phi(.)$:

$$\hat{y}_{nrop}(k) = \Phi(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1\text{:}k\text{-}n), u(k\text{-}1\text{:}k\text{-}n), y(k\text{-}1), K_n) \quad \text{Eq. 4.6}$$

For the case of a SISO system, without loss of generality, the NROP predictor proposed in this work obeys Eq. 4.7, and the general architecture is depicted in Fig. 4.3.

$$\hat{y}_{nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1), \ldots, \hat{y}_{nrop}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) + K_n\, r_e(k\text{-}1) \quad \text{Eq. 4.7}$$

$$r_e(k\text{-}1) = y(k\text{-}1) - \hat{y}_{nrop}(k\text{-}1)$$

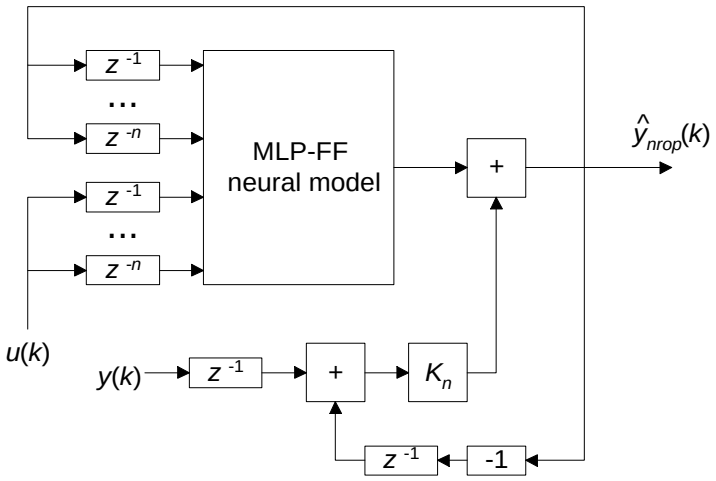


Fig. 4.3 - Architecture of neural recurrent output predictor (NROP).

Each neural network $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W},\ldots)$ is trained off-line, based on the data captured in closed-loop operation, according to a one-step ahead predictor neural model (Eq. 4.4), by minimization of the output prediction error.

The neural predictor (NROP) proposed incorporates two embedded mechanisms: a parallel model (Eq. 4.5) acting as a prediction mechanism, and the term $K_n\, r_e(k\text{-}1)$ acting as a correction mechanism, where the predictor gain K_n has an important role. The output residual is given by $r_e(k) = y(k) - \hat{y}_{nrop}(k)$. This predictor structure is similar to the operation of the classical state observers (Luenberger observer and Kalman filter): a time update ("predict"), and a measurement update ("correct").

This approach can be extended to deal with MIMO systems. Assuming that a MIMO system is decomposed in a set of MISO systems, a set of neural output predictors (NROP) based on MISO system models can be designed to predict each system output.

4.3.5 Training of Neural Models for the NROP Predictor

In order to implement nonlinear models based on neural networks, different neural networks (NN) can be used according to the type of activation functions: radial basis function RBF-NN and multi-layer perceptron MLP-NN (Hagan, et al., 1995). RBF neural networks have the

following features: a local response due to the Gaussian activation function, a high number of neurons in the hidden layer, and are fast to train (an advantage for on-line algorithms). MLP neural networks have the following features: a global response due to the sigmoid activation function, a lower number of neurons in the hidden layer, and are slower to train.

The MLP neural networks have been chosen to be used as nonlinear models in the neural NROP structures proposed in this dissertation, since they have global response and can be trained off-line. A nonlinear NARX neural model with three layers is sufficient for nonlinear dynamic modeling (Haykin, 1994 ; Hagan, et al., 1995): one input layer, one hidden layer (sigmoid), and one output layer (linear).

In this work, the Levenberg-Marquardt optimization algorithm was used for the training of the neural networks (Hagan, et al., 1995).

For the MLP neural networks, the optimal number of neurons in the hidden layer is still an open question in the neural networks scientific community. Theoretically, the number of neurons in the hidden layer is a function of the complexity of the system dynamics. In this work, an empirical rule is suggested to choose the number of neurons in the hidden layer for the case of neural predictor models: "must be approximately equal to the number of inputs", i.e., the number of elements of the regression vector $[y(k\text{-}1) \ldots y(k\text{-}n)\ u(k\text{-}1) \ldots u(k\text{-}n)]$, since the model order must reflect the complexity of the system dynamics.

4.3.6 Dynamic Analysis of Neural Predictor NROP

In Eq. 4.7 for the neural predictor NROP, the gain K_n is a design parameter. The value of the gain must guarantee a stable dynamics, and a small residual. For a given operating condition, the main goal of the NROP is to guarantee that the residual $r_e(k) = y(k) - \hat{y}_{nrop}(k)$ be small. According to (Eq. 4.7), the residual is given by:

$$r_e(k\text{-}1) = \left(\frac{1}{K_n}\right)\left(\hat{y}_{nrop}(k) - NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1),\ldots, \hat{y}_{nrop}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n))\right) \quad \text{Eq. 4.8}$$

Observing Eq. 4.8, theoretically it can be asserted that: "a small residual $r_e(k\text{-}1)$ of the NROP can be achieved using a high gain, unless the residual $\hat{y}_{nrop}(k) - NN_{\{a\text{-}b\text{-}c\}}(.)$ of the parallel model is small enough". In practice, high values for the gain, K_n, usually produce some oscillations on the output signal, $\hat{y}_{nrop}(k)$. So, a practical rule is: "the gain must be low". In

many experimental tests, typical empirical values obtained for the FDD purposes are, usually, in the range]0; 1[.

Stability plays a very important role in the control theory. Also in system identification one must sometimes deal with the stability issue. It does not, however, play the same vital role as in the control theory. In a neural network (NN) model with only bounded activation functions, such as hyperbolic tangent sigmoids, the output is always bounded; this is the case of the neural structures used in this dissertation.

For linear discrete time invariant systems it is well-known that bounded-input bounded-output (BIBO) and asymptotic stability are obtained if the eigenvalues of the system matrix (corresponding to the poles of the transfer function) are strictly inside the stability area, i.e., the unit circle. For the case of time varying nonlinear systems and models, such as neural network models with time-varying inputs, the analysis is much more complicated.

Under the assumption that the time-variation is sufficiently slow, the stability of the neural network or the control system can be established by evaluating the eigenvalues of the linearized system. In heuristic terms, in a stable system the signals should not "explode" and deviate to far from its reference signals. Typically stability problems will be that the system output signals go into excessive oscillations instead of being smooth, or the signals will saturate (Norgaard, et al., 2003).

Next, a new theorem related to the convergence of the neural recurrent output predictor (NROP) is proposed in this work. The underlying ideas are inspired on the heuristic notion of stability previously described.

Theorem 4. Convergence of neural recurrent output predictor (NROP).

For a given SISO (or MISO) system, the output $\hat{y}_{nrop}(k)$ of the neural recurrent output predictor (NROP) in the form

$$\hat{y}_{nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1),\ldots, \hat{y}_{nrop}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) + K_n\, r_e(k\text{-}1) \qquad \text{Eq. 4.9}$$

converges to the output process signal $y(k)$, if the embedded neural model $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W},\ldots)$ is a good one step ahead predictor, i.e., the residual $r_e(k)$ is small enough. The predictor gain K_n allows an adjustment of the predictor output error (residual) $r_e(k) = y(k) - \hat{y}_{nrop}(k)$.

■

Qualitative Reasoning for Theorem 4. If the embedded neural model $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W},\ldots)$ is a good one step ahead predictor then a stable behaviour can be guaranteed for data similar to the one used in the training phase. For these conditions the equation is valid:

$$\hat{y}_{nrop}(k) \approx NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1), \ldots, \hat{y}_{nrop}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) \,. \qquad \text{Eq. 4.10}$$

If Eq. 4.10 is valid, and taking into account Eq. 4.9, then the predictor output error $r_e(k) = y(k) - \hat{y}_{nrop}(k)$ is approximately zero assuming $K_n \neq 0$, i.e, $r_e(k) \cong 0$. Consequently, the NROP predictor output converges to the ouput process model since the expression is valid $\hat{y}_{nrop}(k) \approx y(k)$.

■

Since in practice there is always some kind of non modeled dynamics, the neural model does not perfectly models the plant behaviour, and this causes a non-zero small residual $r_e(k) = y(k) - \hat{y}_{nrop}(k) \neq 0$. The predictor gain ($K_n$, design parameter) allows the adjustment of the amplitude of output residual signal according to Eq. 4.8. Some experiments performed shown that the gain K_n must be small, approximately in the range]0; 1[, in order to guarantee stability and a small residual.

There are three main advantages of the new neural recurrent output predictor (NROP) proposed in this work: a) the training is done off-line; b) only input-output data is required; c) it can be applied in real-time FDD applications for SISO or MIMO nonlinear systems.

4.4 FDD Approach based on a Bank of NROP Predictors

4.4.1 Introduction

For nonlinear systems, some investigations have been carried out related to the application of a bank of classical extended Kalman filters (EKF) to FDD purposes. The main difficulties of this type of approach are: a) obtaining an accurate process model; b) the estimation of the process noise matrix. If these two requirements are not satisfied, the filter can diverge and the

stability is not achieved. These reasons have motivated researchers to develop nonlinear observers based on intelligent (neural and fuzzy) techniques (Zhou & Bennett, 1997; Isermann, 1997; Patton, et al., 2000; Chen & Patton, 1999), and nonlinear neural output predictors (Palma, et al., 2004a; Palma, et al., 2005b).

The neural FDD approach proposed here is carried out only by the evaluation of the plant output $y(k)$, control input $u(k)$, and delayed versions of these signals. Therefore, the FDD approach does not depend directly on the order of the plant. The embedded neural model must be a good output predictor model for the plant dynamics.

An FDD approach based on a bank of neural recurrent output predictors (NROP) described in section 4.3 is presented later. The main idea is to have a NROP tuned to each faulty situation, and one for the nominal operation. For FDD purposes, the NROP residuals (output predictors errors) are used as output residual generators. When a fault occurs, the NROP tuned to this faulty case must present the lower residual $r_e(k) = y(k) - \hat{y}_{nrop}(k)$ compared to the other NROP residuals.

4.4.2 The FDD-NROP Approach

The main question to be solved in this section is formulated next, assuming a nonlinear process and a nonlinear NARX model.

Problem 4. Given a continuous time nonlinear SISO (or MISO) dynamic system S modeled by a black-box nonlinear neural NARX model M_{XNN} in discrete time, find a methodology for fault detection and diagnosis (FDD) in real-time operation.

The main idea used to solve this problem was inspired by the NROP predictor proposed in this work, where for each fault an embedded neural model must be identified via off-line training and tuned by adjusting the gain K_n. To solve this problem a bank of NROP predictors, each one given by Eq. 4.11, as described in the last section, is proposed according to the architecture (Fig. 4.4). Each NROP is labeled as $\Psi(.)$, its architecture is depicted in Fig. 4.3, with input data $y(k)$ and $u(k)$, and output $\hat{y}_{nrop}(k)$.

$$\hat{y}_{nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1), \ldots, \hat{y}_{nrop}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) + K_n\, r_e(k\text{-}1) \qquad \text{Eq. 4.11}$$

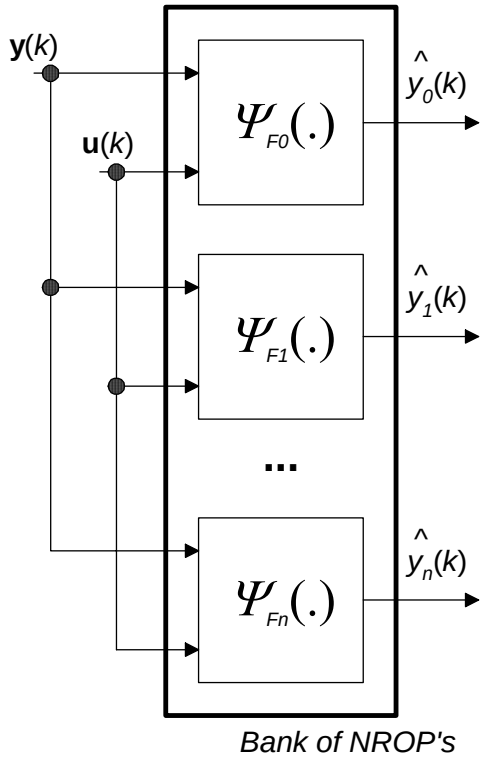


Fig. 4.4 - Architecture of the bank of NROP predictors.

Based on the input-output available data, for the nominal operation conditions and for each faulty case, neural output predictor models $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, y(k\text{-}1), \ldots, u(k\text{-}1), \ldots)$ are trained off-line and saved in neural structures. Each neural model is used as an embedded model tuned to each faulty situation in a NROP predictor $\Psi_{Fi}(.)$. The NROP predictor $\Psi_{F0}(.)$ is the one tuned for the nominal operating conditions (the case of fault F_0), and $\Psi_{Fi}(.)$ is the one tuned for fault F_i. This means that for detection of n faults, n+1 NROP predictors are required.

Fault Detection (FDE) Approach. For the fault detection task, a residual generator can be obtained directly from the residual (output prediction error) of the NROP predictor tuned for the nominal operation, $\Psi_{F0}(.)$. The FDE approach can use a threshold based on a three sigma limit, or a more robust method based, for example, on an adaptive CUSUM algorithm (Gustafsson, 2001; Palma, et al., 2004b). Here, a three sigma limit is used.

A fault alarm is based on the square of prediction error (SPE), $q(k) \in \Re^{1 \times 1}$, computed based on the residual $\mathbf{r}_0(k) = \mathbf{y}(k) - \hat{\mathbf{y}}_0(k)$, with $\mathbf{r}_0(k) \in \Re^{w \times 1}$. For a sliding window of length w, the SPE signal is given by

$$q(k) = \mathbf{r}_0(k\text{-}w\text{+}1{:}k)^{\mathrm{T}}\, \mathbf{r}_0(k\text{-}w\text{+}1{:}k). \qquad \text{Eq. 4.12}$$

A fault alarm is generated if the SPE, $q(k)$, exceeds the threshold values **h**, i.e., $a_m(k) = 1$. Otherwise, no fault alarm occurs. For the three sigma limit method, the thresholds are given by $[\mu - 3\sigma; \mu + 3\sigma]$, where μ is the mean value and σ is the standard deviation of the SPE signal computed using data captured for nominal operation. The fault detection signal $f_d(k)$ is obtained by low pass filtering the fault alarm signal $a_m(k)$ and by thresholding.

Fault Diagnosis (FDG) Approach. The proposed fault diagnosis approach depends on the performance of each NROP predictor. If the FDD system is designed to isolate only two faults, then a bank of three neural observers is required. One predictor $\Psi_{F0}(.)$ tuned to the nominal operating region (without faults, F_0), one predictor $\Psi_{F1}(.)$ tuned to the fault F_1, and another $\Psi_{F2}(.)$ tuned for the fault F_2. If fault F_1 occurs, then the associated residual $r_1(k) = y(k) - \hat{y}_1(k)$ of NROP predictor $\Psi_{F1}(.)$ will tend to a small value near zero, and the other residuals will tend to a non-zero value or even present some oscillations. The square of prediction error (SPE) is computed based on each residual, and after low pass filtering $H_{lp}(z, \lambda)$ the isolated fault is determined by the lower square of prediction error.

The general architecture (Fig. 4.5) based on the ideas described before related to the FDD approach based on a bank of NROP predictors is presented next.

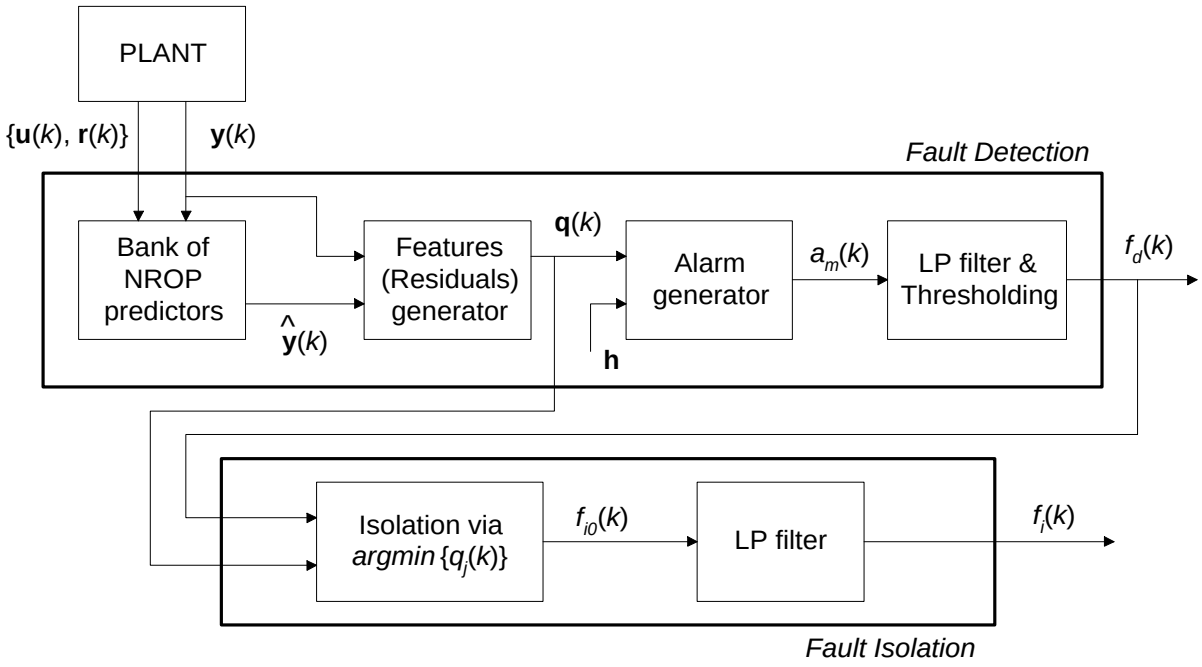


Fig. 4.5 - Architecture of the FDD approach based on a bank of NROP predictors.

4.4.3 Algorithm and Example

Next the algorithm used for fault detection and diagnosis based on a bank of neural recurrent output predictors (NROP) is presented. Without loss of generality, the algorithm considers the case of a SISO system. It is straightforward to extend the algorithm for the case of MISO systems.

Algorithm 6. Fault detection and diagnosis based on a bank of neural recurrent output predictors (FDD-NROP).

In off-line operation, the following tasks must be executed:

a. For each fault F_i of the set $F = \{F_0, F_1, \ldots, F_n\}$ proceed with the training of the neural network output predictor model $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}_i, y(k\text{-}1), \ldots, u(k\text{-}1), \ldots)$ tuned to the respective fault F_i, using the Levenberg-Marquardt optimization algorithm. A set of neural predictor models is obtained, where each predictor is characterized by a different weight matrix $\mathbf{W}_i$.

b. Initialize the output $\hat{y}_{nrop}(k)$ of the NROP predictor with the process output $y(k)$ under nominal conditions during the start-up.

c. Determine the thresholds and the low pass filters parameters, in order to obtain a desired commitment between rate of false alarms, rate of missed fault detections, and detection and isolation delays. Different sets of experimental nominal data must be used to compute and validate the thresholds and filters parameters.

Each time instant k, the following steps must be executed on-line:

1. Sample the process output signal $y(k)$.

2. Compute the output predictor signal, for each faulty case, based on the neural recurrent output predictor (NROP) approach proposed in the work. The output predictor signal, for the NROP tuned to fault F_i, and denoted by $\Psi_{Fi}(.)$, is given by:

$$\hat{y}_{nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}_i, \hat{y}_{nrop}(k\text{-}1),\ldots, \hat{y}_{nrop}(k\text{-}n_a), u(k\text{-}n_d),\ldots, u(k\text{-}n_c)) + K_{n\_i}\, r_e(k\text{-}1) \quad \text{Eq. 4.13}$$

3. Compute the output residual signal, $r_i(k) = y(k) - \hat{y}_{nrop}(k)$, for each NROP predictor. A set of residuals $R = \{r_0(k), r_1(k), \ldots, r_n(k)\}$ is obtained for the bank of NROP predictors.

4. If the SPE, $q(k)$, associated with the residual of the nominal NROP predictor $\Psi_{F0}(.)$, for a sliding window, exceeds the thresholds, then a fault alarm $a_m(k)$ is generated. It is assumed that the threshold is computed using a three sigma limit approach.

5. Compute the fault detection signal $f_d(k)$ by low pass filtering via $H_{lp}(z, \lambda)$ the alarm signal $a_m(k)$, and by thresholding.

6. If a fault is detected, i.e. $f_d(k) = 1$, compute the signal $f_{i0}(k)$ according to the smallest SPE error in the set $Q_n = \{q_0(k), q_1(k), \ldots, q_n(k)\}$:

$$f_{i0}(k) = argmin_j \{q_j(k); j = 0, 1, \ldots, n\} \qquad \text{Eq. 4.14}$$

7. The fault isolation $f_i(k)$ is obtained by low pass filtering $f_{i0}(k)$ using a LP filter $H_{lp}(z, \lambda)$ (described in section 2.4.4).

■

Next, an example of application of the NROP fault detection and diagnosis methodology is presented.

Example 8. Fault detection and diagnosis approach based on a bank of NROP predictors applied to a DC motor model.

In this example, the proposed approach based on a bank of NROP predictors is applied to a DC motor model.

The continuous time DC motor model (described in more detail in section 5.3.2) is expressed by the transfer function:

$$G_m(s) = \frac{\omega_r(s)}{u_a(s)} = \frac{K_m}{(L\,J)\,s^2 + (L\,K_f + R\,J)\,s + (R\,K_f + K_m\,K_b)}. \qquad \text{Eq. 4.15}$$

The general architecture of the plant model under study is depicted in Fig. 4.6.

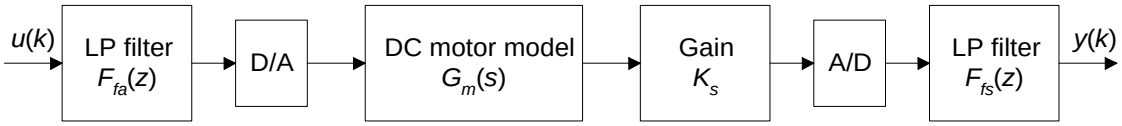


Fig. 4.6 - DC motor model with low pass filters.

In this example, parametric faults on the sensor gain K_s are considered. The nominal value of K_s is 4.1. The set of faults is denoted by $F = \{F_0, F_1, F_2, F_3\}$. The nominal operation is denoted by fault F_0, and the other faults correspond to the multiplication of the nominal sensor gain by the set of values {0.1, 0.3, 5}.

A bank of four NROP predictors, each one tuned to a specific fault, has been constructed and used in this example. All the NROP predictors have the same gain $K_n = 0.1$.

The neural NARX(…) structures used in the predictor models use a time horizon equivalent to an ARX($n_a = 3$, $n_b = 1$, $n_d = 2$) model relating the output signal $y(k)$ and the input signal $u(k)$. For each neural predictor model $NN_{\{a\text{-}b\text{-}c\}}(\ldots)$, the number of neurons in each layer is $a = 4$; $b = 4$; $c = 1$. The transfer functions for the hidden layer and for the output layer used are, respectively, the hyperbolic tangent sigmoid (tansig) and the linear (purelin). The neural predictor models have been trained off-line using the Levenberg-Marquardt backpropagation (LMBP) algorithm.

The system is controlled by a linear discrete PI controller (Astrom & Hagglund, 1988). After tuning via a pole-placement approach, the controller gains obtained are $K_p = 2.56$ and $T_i = 2.02$ s. The sampling time is $T_s = 0.11$ s.

The DC motor model is linear, but the faults considered cause a nonlinear effect (actuator saturation). The next figures show the results obtained for the fault F_1, corresponding to an abrupt decrease of the sensor gain (K_s). The sensor gain changes from the nominal value (4.1) to a faulty value (0.1*4.1 = 0.41). The variance of the sensor noise is 1×10^{-8}.

Fig. 4.7 shows, from top to bottom, the following signals for fault F_1.

The reference signal $r(k)$ with a dither signal with variance 1×10^{-4} added, the output signal $y(k)$, and the input signal $u(k)$. Next, the fault alarm signal $a_m(k)$, the fault detection signal $f_d(k)$ and the fault isolation signal $f_i(k)$ are shown. Here, an analysis of the fault magnitude is not considered, and for that reason the fault analysis signal $f_a(k)$ is zero.

For an experiment of 400 s of duration, the fault occurs at time instant $t_k = 240$ s. The detection delay is 0.5 s, and the isolation delay is nearly 6 s.

As observed, some false alarms occur. The low pass filtering of the alarm signal $a_m(k)$ avoids the activation of the fault detection signal $f_d(k)$. The low pass filtering plays an important role in most FDD approaches, since it renders the system more robust with respect to false alarms.

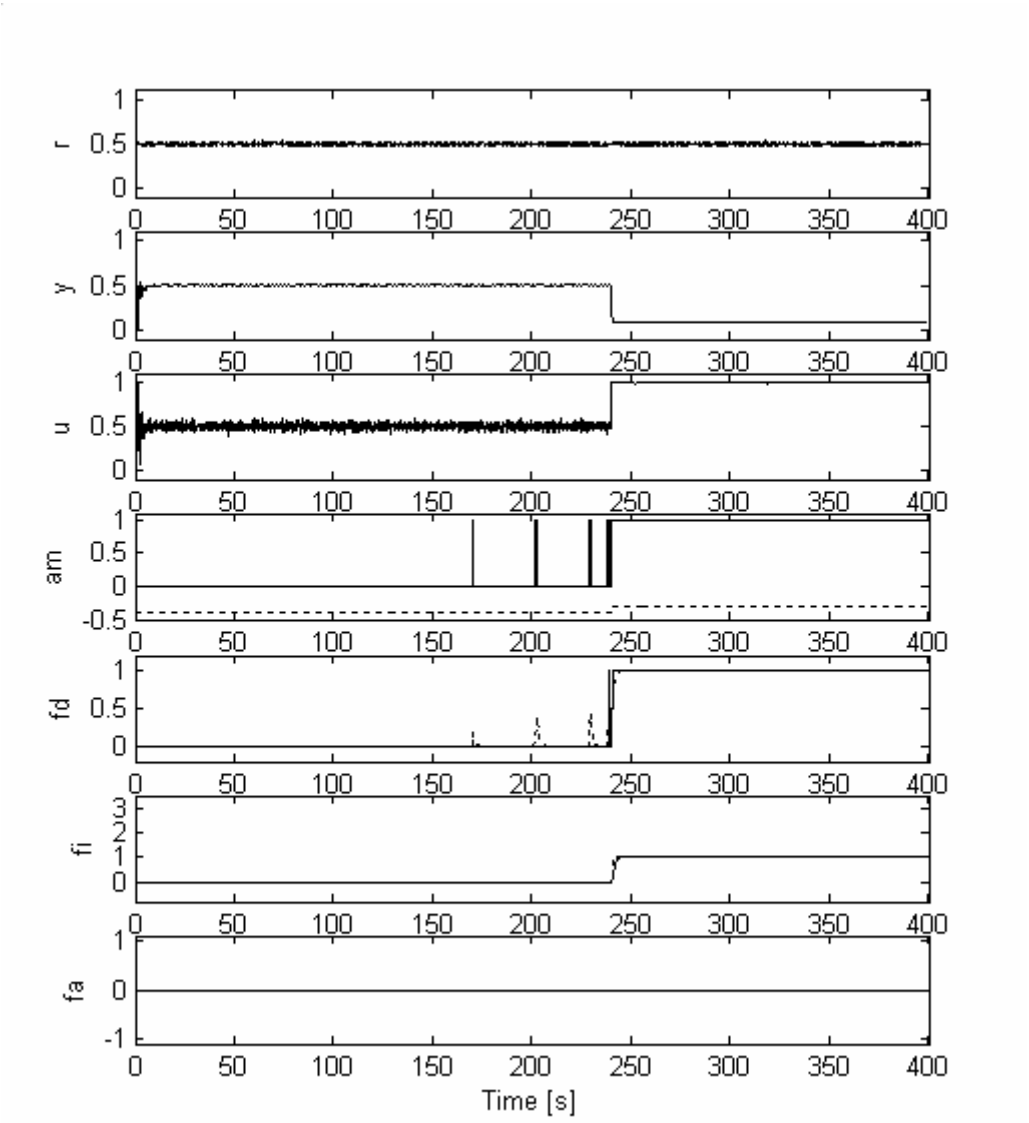


Fig. 4.7 - Signals for FDD based on NROP applied to a DC motor model.

The residuals obtained from the bank of NROP predictors are used for fault detection and isolation. The first graph of Fig. 4.8 shows the reference signal $r(k)$ (red line), the output signal $y(k)$ (blue line) and the input signal $u(k)$ (green line). This fault causes the saturation of the actuator at the maximum value (magnitude around 1). Since the nominal static gain of the plant model is 1, when the fault occurs the output signal tends to the value 0.1. The residual $r_0(k)$ for the nominal NROP predictor is shown next. After, the residuals ($r_i(k)$; i = 1, …, 3) of the NROP predictors tuned to each fault can be seen. The Square of Prediction Error (label rq), $q(k)$, for the nominal predictor used for fault detection is shown next, and also the threshold values. The SPE signal is computed based on a sliding window of length w_s= 1 s, i.e., a sliding window with 9 samples. Finally, the fault isolation signal $f_i(k)$ is depicted. Fault F_1 is isolated since the output residual of the associated NROP predictor is the smallest.

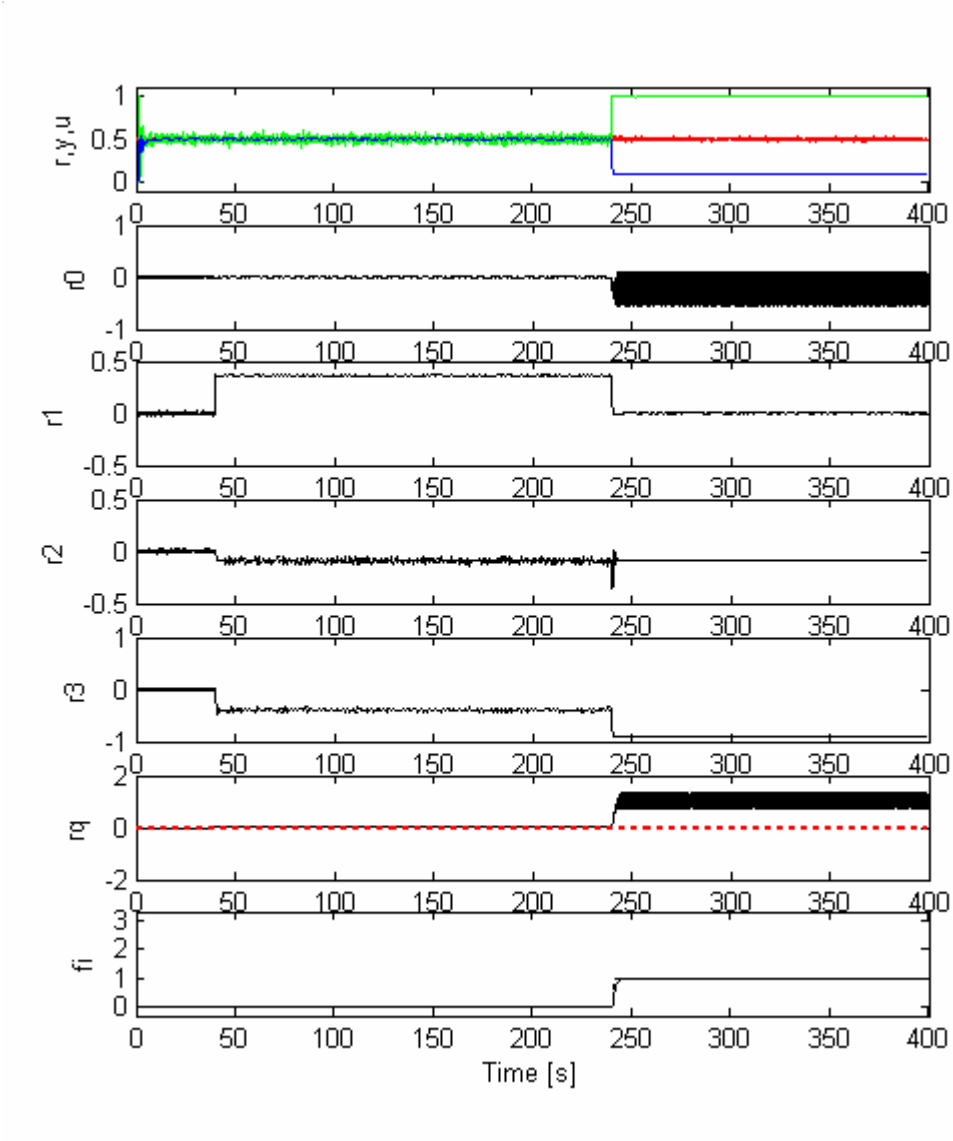


Fig. 4.8 - Output residual signals for NROP predictors.

This FDD approach is appropriate for many faulty cases, including some faulty cases where other approaches fail, such as situations of saturation and oscillatory behaviours. Neural network predictor models are able to capture this type of nonlinear behaviours. In some cases of saturation, the persistent excitation conditions (PEC) are not good enough and this fact causes an increase on the variance of the on-line estimated parameters of the ARX models. In the top graph (label th-yu) of Fig. 4.9, the parameters $\boldsymbol{\theta}(k) = [a_1(k)\ a_2(k)\ b_2(k)]$ of an ARX(2, 1, 2) model $M_{yu}(\boldsymbol{\theta})$ are depicted; the nominal values are given by $[a_1\ a_2\ b_2] = [-1.76\ 7.84\times10^{-1}\ 2.48\times10^{-2}]$. Next (label th-yr), the parameters of an ARX(2, 1, 2) model $M_{yr}(\boldsymbol{\theta})$ are shown; the vector $[a_1\ a_2\ b_2] = [-1.79\ 8.56\times10^{-1}\ 6.18\times10^{-2}]$ shows the nominal values. The last signal (label fas) is the fault activation (trigger) signal. The on-line parameter estimation has been made using the sliding window SW-PCR algorithm.

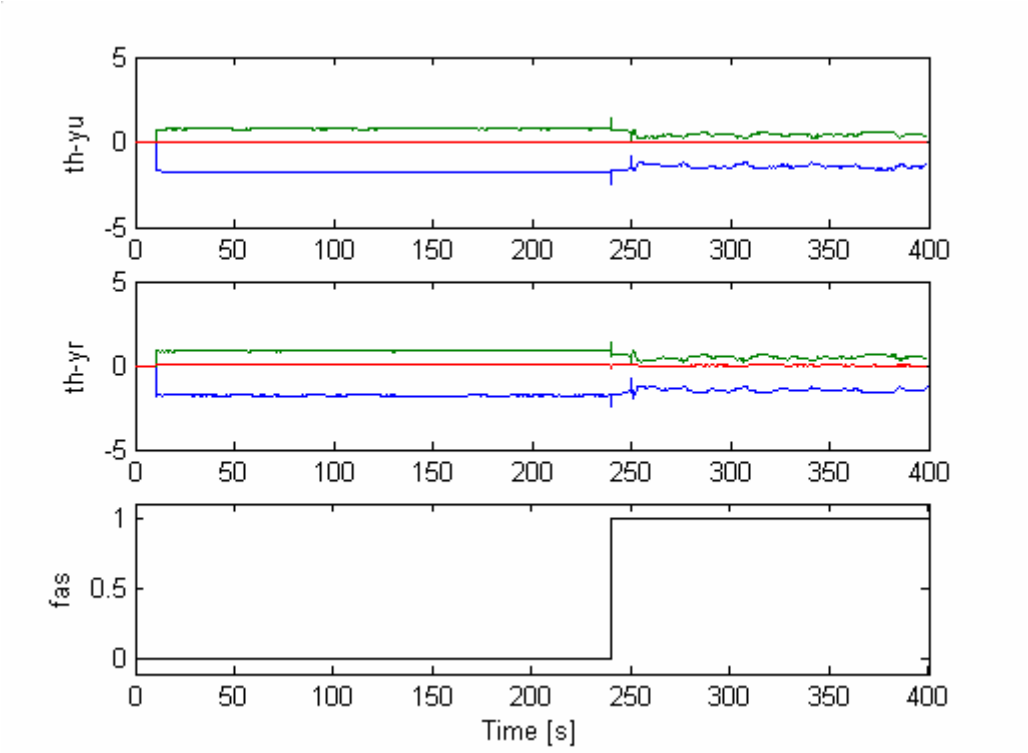


Fig. 4.9 - Parameter estimates in actuator saturation case.

4.5 FDD Approach based on Neural Nonlinear PCA and Neural NNLDA

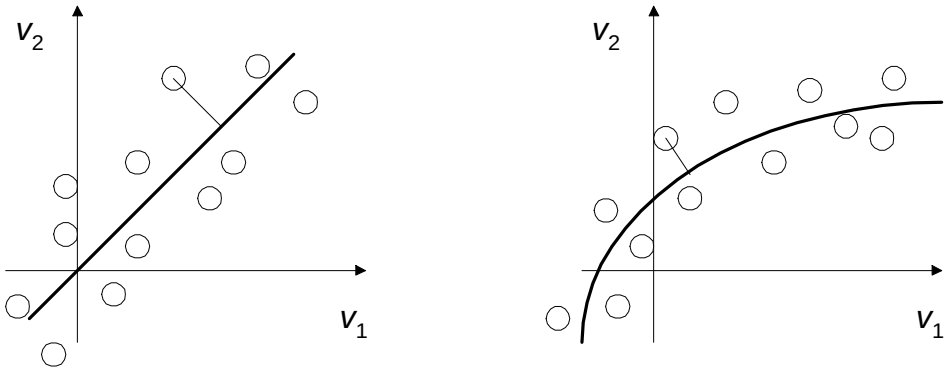
4.5.1 Introduction

The data-driven approaches to fault detection and diagnosis (FDD) are based on models built directly from process data. The great strength of data-driven techniques is their ability to transform the high dimensional data into a lower dimension, in which the important information is captured. This is particularly important in large-scale industrial systems (chemical plants, nuclear plants, etc) that produce a large amount of multivariate data.

The Principal Component Analysis (PCA) described in Chapter 2 is a linear technique, and is the most widely statistical multivariate technique used in industry (Jackson, 2003; Jolliffe, 2002).

In many situations the relations between data are not linear, so the linear PCA techniques are not the most appropriate. For the cases of nonlinear relations in the data, a better solution is the application of nonlinear PCA techniques. The nonlinear principal component analysis

(NLPCA) is a generalization of PCA to nonlinear systems. The two main NLPCA techniques are based on Principal Curves (Hastie, 1984; Harkat, 2003), and on Kramer's Neural NLPCA (Kramer, 1991; Harkat, 2003). The linear PCA technique extracts the linear relations between variables, and the aim of nonlinear PCA is to extract the linear and nonlinear relations, as observed in Fig. 4.10 (Harkat, 2003).



a) Linear PCA. b) Nonlinear PCA.

Fig. 4.10 - Linear PCA and nonlinear PCA.

The nonlinear principal components do not obey a normal distribution. The confidence regions cannot be derived from the normal distribution, but can be obtained using approximations based on kernel functions (Lopes, 2001). The kernel functions can be used as estimators of the probability density functions. A stochastic kernel function is assumed to be independent and identically distributed, thus it can be used as an estimator of the probability density function (PDF) of a stochastic process in discrete time.

There is an extended list of publications related to linear PCA. With respect to nonlinear PCA, the number of publications is reduced. Some publications related to NLPCA applied directly to input-output data with applications to real systems are the papers (Dunia, et al., 1995; Antory, et al., 2004; Antory, et al., 2005; Palma, 2006).

In this section the possibility of application of neural nonlinear PCA to input-output data is discussed, and also to ARX model parameters identified on-line.

A new approach to FDD in nonlinear systems based on neural nonlinear principal component analysis (neural NLPCA) is proposed. Instead of the classical approach where NLPCA is applied to input-output data, here NLPCA is applied to the set of parameters of an adaptive ARX model identified on-line. Fault detection and isolation is performed using neural nonlinear discriminant analysis (NNLDA). Both the two dimensional scores and the SPE

signal are used as features for FDD. This methodology can be applied to SISO or MIMO systems.

4.5.2 Review of Classical and Neural Linear PCA

Classical PCA is a linear technique, based on multivariate statistics, for mapping multidimensional data into lower dimensions with minimal loss of information. Since PCA has been described in many references and also in Chapter 2, only a brief summary of classical linear PCA is given here.

Let $\mathbf{X} \in \Re^{n \times m}$ represent a data matrix (n = number of observations, m = number of variables), in which the observations are mean centered and appropriately scaled. PCA is an optimal factorization of $\mathbf{X}$ into two matrices, $\mathbf{T} \in \Re^{n \times a}$ called the scores matrix, and $\mathbf{P} \in \Re^{m \times a}$ called the loadings matrix, plus a matrix of residuals $\mathbf{E} \in \Re^{n \times m}$:

$$\mathbf{X} = \mathbf{T}\,\mathbf{P}^{\mathrm{T}} + \mathbf{E} \qquad \text{Eq. 4.16}$$

In Eq. 4.16, the number of principal components (factors) is given by a (the number of columns of matrices $\mathbf{T}$ and $\mathbf{P}$). The condition of optimality on the factorization is that the Euclidean norm of the residual matrix, $||\mathbf{E}|| = ||\mathbf{X} - \hat{\mathbf{X}}||$, must be minimized for the given number of principal components, a. To satisfy the criterion, the columns of $\mathbf{P}$ are the eigenvectors corresponding to the a largest eigenvalues of the covariance matrix of $\mathbf{X}$.

PCA can be viewed as a linear mapping of data from $\Re^m$ to $\Re^a$. Taking $\mathbf{P}^{\mathrm{T}}\,\mathbf{P} = \mathbf{I}$, without loss of generality, the mapping has the form:

$$\mathbf{T}_r = \mathbf{X}_r\,\mathbf{P} \qquad \text{Eq. 4.17}$$

In Eq. 4.17, $\mathbf{X}_r$ represents a row of matrix $\mathbf{X}$, a single data vector. The vector $\mathbf{T}_r$ represents the corresponding row of $\mathbf{T}$, or the coordinates of $\mathbf{X}_r$ in the reduced features (scores) space. The loadings $\mathbf{P}$ are the coefficients for the linear transformation. The information lost in this mapping can be assessed by reconstruction of the data by reversing the projection back to $\Re^m$:

$$\hat{\mathbf{X}}_r = \mathbf{T}_r\,\mathbf{P}^{\mathrm{T}} \qquad \text{Eq. 4.18}$$

According to equations Eq. 4.16 and Eq. 4.18, the error vector is given by

$$\mathbf{E}_r = \mathbf{X}_r - \hat{\mathbf{X}}_r. \qquad \text{Eq. 4.19}$$

For a smaller dimension of the scores (features) space, a greater resulting error is obtained.

The linear PCA multivariate statistics technique can also be implemented using neural networks (Kramer, 1991; Diamantaras, 1996). The architecture of the neural network is similar to the one presented in the next sub-section related to nonlinear neural PCA, but only a three layer with linear activation functions is necessary.

4.5.3 Kramer's Neural Nonlinear PCA

The principal curves method and the Kramer's neural NLPCA method are the two main approaches to extend PCA to deal with nonlinear systems (Harkat, 2003). The Kramer's neural approach was used in this work, and will be briefly described. Kramer's NLPCA approach has been developed motivated by the combination of concepts from a data-driven approach (PCA), and from a knowledge-based technique (neural networks).

In Kramer's NLPCA approach, the mapping into the features (scores) space is generalized to allow arbitrary nonlinear functionalities (Kramer, 1991). By analogy to Eq. 4.17, we seek a mapping in the form:

$$\mathbf{T}_r = G(\mathbf{X}_r) \qquad \text{Eq. 4.20}$$

where G is a nonlinear vector function, composed of a individual nonlinear functions; the vector $G = \{G_1, G_2, \ldots, G_a\}$ is analogous to the columns of matrix $\mathbf{P}$, such that $\mathbf{T}_{ri}$ represents the i^{th} element of $\mathbf{T}_r$:

$$\mathbf{T}_{ri} = G_i(\mathbf{X}_r) \qquad \text{Eq. 4.21}$$

By analogy to the linear case, G_1 is referred to as the first nonlinear principal component (or primary nonlinear factor), and G_i is the i^{th} nonlinear principal component of $\mathbf{X}_r$.

The inverse transformation, restoring the original dimensionality of data, analogous to Eq. 4.18, is implemented by a second nonlinear vector function $H = \{H_1, H_2, \ldots, H_m\}$:

$$\hat{\mathbf{X}}_r = H(\mathbf{T}_r) \qquad \text{Eq. 4.22}$$

The loss of information is again measured by $\mathbf{E}_r = \mathbf{X}_r - \hat{\mathbf{X}}_r$, and the nonlinear functions G and H are selected to minimize the error matrix, $||\mathbf{E}|| = ||\mathbf{X} - \hat{\mathbf{X}}||$.

Kramer (1991) showed that an auto-associative neural network (NN), with the architecture depicted in Fig. 4.11, is able to implement the nonlinear transformations (mapping G and de-mapping H) required by the nonlinear PCA method. This architecture allows the simultaneous determination of a nonlinear principal components (factors). The extraction of nonlinear principal components can be done using two algorithms (Karkat, 2003). For the series algorithm, each principal component is obtained one at a time, using only one neuron in the bottleneck layer. The parallel algorithm, which is used in this work, extracts the principal components (#PC = "a") at once, using "a" neurons in the bottleneck layer.

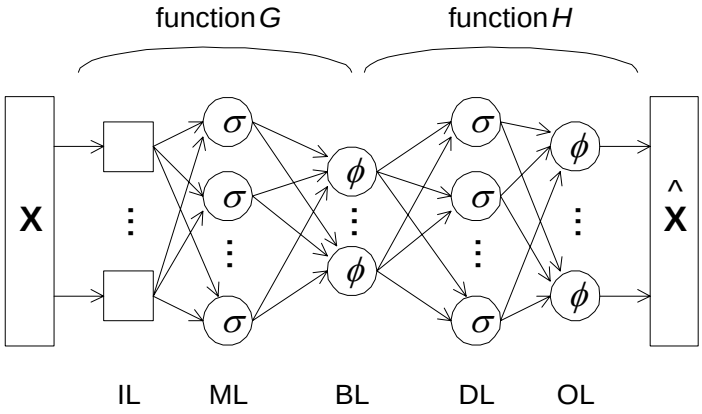


Fig. 4.11 - Schematic diagram of an auto-associative neural network used in NLPCA.

As it can be observed, the auto-associative neural network architecture proposed by Kramer consists of five layers: an input (buffer) layer (IL), a mapping layer (ML), a bottleneck layer (BL), a de-mapping layer (DL), and an output layer (OL). The training of the NN can be performed using the Levenberg Marquardt optimization algorithm, or other methods based on the back-propagation algorithm. To implement a neural nonlinear PCA approach, the activation functions (σ) of the neurons in the mapping and de-mapping layers are usually of the type sigmoid (tansig, or logsig), and usually linear functions (ϕ) are used in the bottleneck and output layers. The input layer receives the measured process variables (or related variables) expressed by matrix $\mathbf{X}$. For a given observation, vector $\mathbf{X}_r$, the values of the

nonlinear scores variables are given by the output of the bottleneck layer, $\mathbf{T}_r = G(\mathbf{X}_r)$. The residual vector of the reconstruction is given by $\mathbf{E}_r = \mathbf{X}_r - \hat{\mathbf{X}}_r = \mathbf{X}_r - H(G(\mathbf{X}_r))$.

In the linear PCA approach it is possible to quantify the explained variance by a certain number of principal components, as described in section 3.5.2 by Eq. 3.36. For the nonlinear PCA approach, it is also possible to compute the amount of explained variance by the nonlinear principal components, and this value is given in percentage by

$$E_{\sigma 2}(a)\ (\%) = 100 \times \left(1 - \frac{tr(\mathbf{E}^T \mathbf{E})}{tr(\mathbf{X}^T \mathbf{X})}\right) \qquad \text{Eq. 4.23}$$

where $\mathbf{X} \in \Re^{n \times m}$ is the data matrix, $\mathbf{E} \in \Re^{n \times m}$ is the residual matrix given by $\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}}$ obtained after the neural network training, and the function $tr(.)$ gives the trace of a matrix. The predicted data by the NLPCA model is given by $\hat{\mathbf{X}}$.

There is no definite method for deciding a priori the dimensions of the mapping and de-mapping layers (henceforth, collectively referred to as the mapping layers). The number of neurons in the mapping layers is related to the complexity of the nonlinear functions that can be generated by the neural network.

For the nonlinear optimization to work well the input variables must be standardized, i.e. auto-scaled. Proper scaling of the input data is essential to avoid having the nonlinear optimization algorithm searching for parameters with a wide range of magnitudes. Due to the presence of multiple minima in the cost function, the NLPCA is generally less stable than linear PCA (Hsieh, 2001).

If the goal is to implement a neural linear PCA approach, then all activation functions must be linear (ϕ), and only a three layer neural network is necessary (Kramer, 1991).

4.5.4 FDD Classical Approach Based on Neural Nonlinear PCA

Traditionally, the nonlinear principal component analysis (NLPCA) approach is applied to input-output data signals in multivariable nonlinear systems (Diamantaras, 1996; Harkat, 2003; Antory, et al., 2005). In MIMO plants, the number of input and output signals is, in many situations, greater than ten variables (Antory, et al., 2004). The classical neural NLPCA approach can also be applied to SISO systems using Kramer's neural nonlinear PCA approach (Palma, et al., 2006). In this section, the main difference relative to other previous works in this area is the application of neural NLPCA to a SISO plant, instead of MIMO systems. Next

the traditional approach for FDD based on neural nonlinear PCA is described concisely, and an architecture is proposed in Fig. 4.12. The approach is described for SISO systems, without loss of generality. The general problem to be solved can be formulated as follows.

Problem 5. Given a nonlinear continuous time system, find a method for fault detection and diagnosis based on input-output data, using nonlinear principal component analysis (NLPCA).

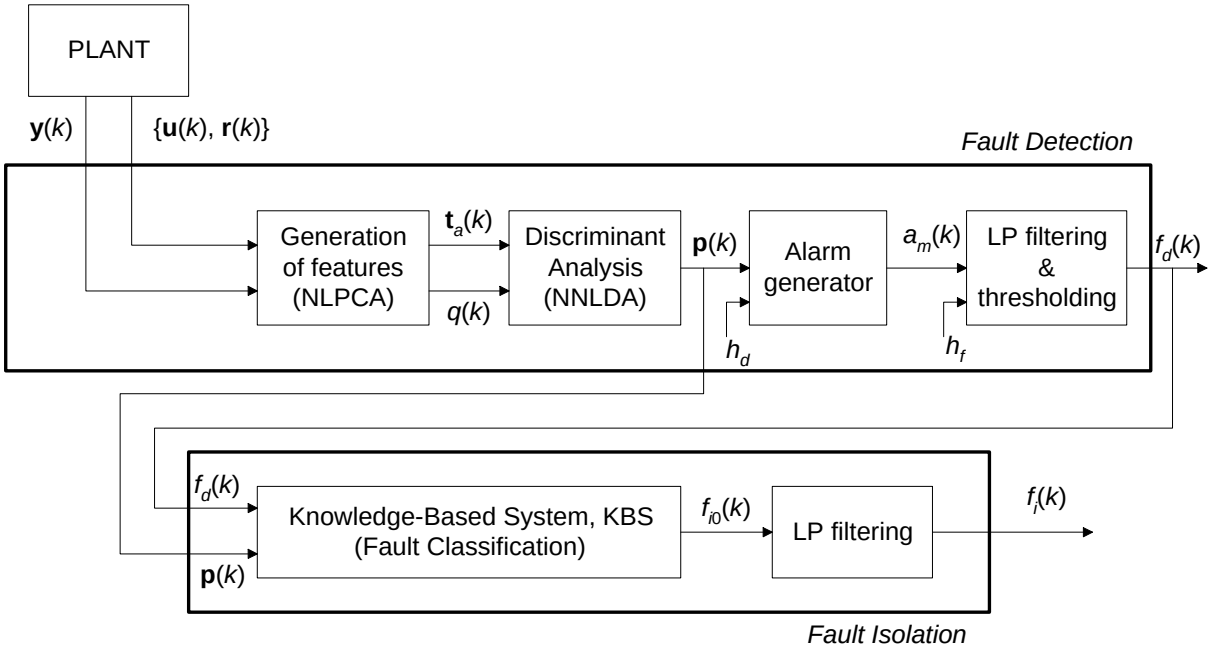


Fig. 4.12 - FDD architecture based on classical neural NLPCA.

The FDD methodology proposed here follows the architecture depicted in Fig. 4.12, and is described afterwards. The main idea is to detect and diagnose faults in the scores space and residual space of the NLPCA model. Many FDD methods based on NLPCA are founded on geometrical approaches and traditional statistical approaches. Here an FDD approach based on a pattern classification method using a neural nonlinear discriminant analysis is proposed, similar to the one described in section 3.3.2.

First, a neural NLPCA model is constructed for the nominal operating region, using input-output signals. In the approach presented here, both the two dimensional scores and the SPE error are used for fault detection and diagnosis. The neural nonlinear discriminant analysis (NNLDA) is used for FDD based on a data pattern given by the scores $t_1(k)$ $t_2(k)$ and the square of prediction error $q(k)$ (SPE, obtained from the residual) represented by the vector $[t_1(k)\ t_2(k)\ q(k)]$. Each fault is classified to a certain class, i.e., neural NNLDA maps an input (pattern) into an output (class).

The NLPCA model can be a local model around a certain reference (set-point), or a global model for the whole range of set-points. The auto-associative neural network is trained off-line with the output data matrix equal to the input data matrix, i.e, $\mathbf{X}_{out} = \mathbf{X}_{in} = \mathbf{X}$ with $\mathbf{X}$ representing the data matrix for nominal operation. The data matrix $\mathbf{X}$, used for building the NLPCA model, has row vectors in the form of $\mathbf{X}_r(k) = [y(k)\ y(k\text{-}1)\ y(k\text{-}2)\ u(k\text{-}2)]$ for the case of a NARX(2, 1, 2) model; k is the discrete-time, and $y(k)$ and $u(k)$ are, respectively, the output and the input signals. All the data is captured in closed-loop, and with a persistent excitation signal added to the reference signal (set-point). For this data matrix, the number of neurons in each layer can be given by: IL(#4), ML(#5), BL(#2), DL(#5), OL(#4). The data signals are compressed to two dimensions (scores space), since the number of neurons in the bottleneck layer (BL) is 2. The number of neurons in the input and the output layer is equal to the number of input signals, since the input data is equal to the output data in the training phase of the neural network. The auto-associative neural network is trained using the Levenberg-Marquardt optimization algorithm (Hagan, et al., 1995). The activation functions used are the hyperbolic tangent sigmoid ("tansig") for the nonlinear function σ, and the linear function ("purelin") for the function ϕ.

In on-line operation, the on-line vector (pattern) $v(k) = [t_1(k)\ t_2(k)\ q(k)]$ is obtained from the NLPCA model, and the neural classifier based on NNLDA assigns a pattern to the respective fault class F_i.

The FDD architecture depicted in Fig. 4.12 is similar to the architecture previously proposed in Chapter 3 and depicted in Fig. 3.5, and the algorithms are consequently also similar. That is the reason why it is not described here.

Afterwards, an example of application of this type of FDD approach based on input-output data is presented.

Example 9. FDD neural NLPCA-NNLDA approach using input-output data applied to a SISO system.

In this example, the classic FDD neural NLPCA approach described using input-output data is applied to a real nonlinear DC motor (the DCM-RA setup). The FDD methodology obeys the architecture depicted in Fig. 4.12.

The setup is a nonlinear system, and exhibits a small time variant behaviour. An anti-aliasing analog filter with cutoff frequency $f_{c1} = 1$ Hz has been used at the system output. More

detailed information about the DCM-RA motor setup can be found in section 5.3.3. An IIR low pass digital filter, with transfer function $H_{lp}(z, \lambda)$ described in section 2.4.4, has been introduced at the system output, in order to reduce the system bandwidth. A value of $\lambda = 0.7$ was selected. The sampling period used is $T_s = 0.11$ s.

Here, in off-line, a neural NLPCA model has been created with standardized data, i.e., data auto-scaled. The data has been auto-scaled to guarantee approximate data with mean zero and unitary variance for the nominal operating region (a set point around 0.7). For this case, data with mean zero and unitary variance is not perfectly achieved since the system is nonlinear and time variant. For the nonlinear PCA approach, the input data is given by the row vector $\mathbf{X}_r(:,k) = [y(k)\ y(k\text{-}1)\ y(k\text{-}2)\ u(k\text{-}2)]$, assuming a NARX(2, 1, 2) model. In the training phase of the auto-associative neural network, a sum of squared errors (SSE) around 2×10^{-3} was obtained at the end of 200 epochs. The number of neurons in each layer is given by, respectively, {4, 5, 2, 5, 4}.

The row input pattern vector given by the scores and the SPE, $[t_1(k)\ t_2(k)\ q(k)]$, is classified by a neural network implementing the nonlinear discriminant analysis (NNLDA). This type of NNLDA approach, described in section 3.3.2, has been also used in Example 1. Assuming the existence of four faults, the fault F_0 (nominal operation) is classified as belonging to the class F0 ([1 0 0 0]), to fault F_1 corresponds the class F1 ([0 1 0 0]), etc. In the training phase of the neural network, a SSE error around 4×10^{-3} was obtained at the end of 500 epochs. The number of neurons in each layer are given by, respectively, {3, 10, 3, 4}.

Four faults are considered in this example. The nominal operating region for a set point around 0.7 is termed the fault F_0. Faults F_1 and F_2 correspond, respectively, to a blocking of sensor at the values 0.5 and 0.8. Finally, fault F_3 corresponds to a change to a critical operating region (a set-point around 0.4) outside the nominal region, where the speed of the motor is near a minimum acceptable value.

The fault clusters, in the scores space, are shown in Fig. 4.13. It can be observed that the cluster F0 associated with the nominal operation region is not centered on the point (0, 0), since the system is nonlinear and time variant, and also the data does not obey a perfectly normal distribution. The dispersion observed in clusters F0 and F3 are due to small oscillations in the system caused by the existence of a dither signal added to the reference signal. This cluster information from the scores and also from the square of prediction error (SPE) signal of the residual are used as patterns for training the discriminant neural network of the NNLDA approach. This neural network acts as a pattern classifier.

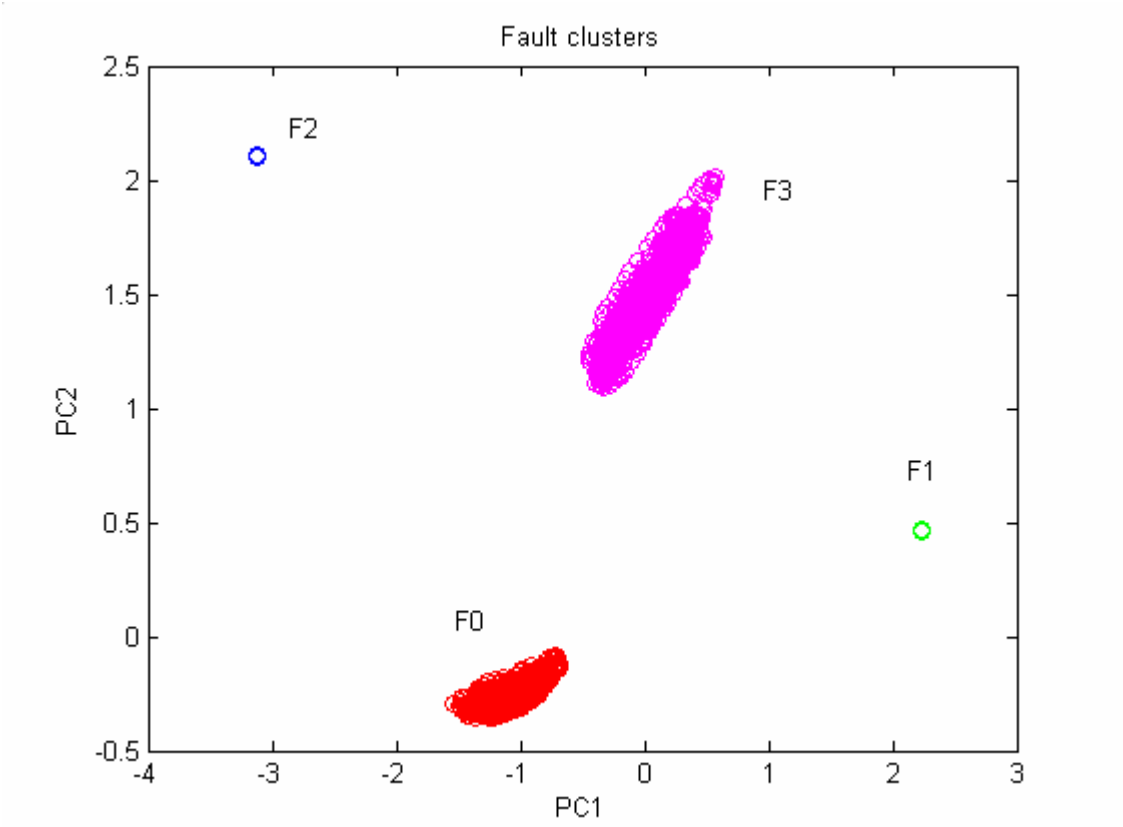


Fig. 4.13 - Fault clusters in scores space.

In this example, results are presented which have been obtained with the fault F_1, corresponding to a sensor blocked at the value 0.5. The fault F_1 considered in this example is a blocking of a sensor at a certain value (0.5), making the plant unobservable because the information link between the plant and the controller is broken. An experiment, under closed-loop control, of duration 400 s was performed. An adaptive optimal linear quadratic Gaussian controller (LQGC) is used to control the speed of the DC motor, based on an input-output ARX(2, 1, 2) model identified on-line, and using a design parameter of $r_0 = 0.4$ to adjust the closed-loop dynamics (Lewis, 1996). A brief description of the LQG controller can be found in section 5.4. At the start-up, a PI controller with gains $K_p = 1$ and $T_i = 2$ s is used, and after the supervisor switch to the adaptive LQG controller.

Fig. 4.14 shows for fault F_1, from top to bottom, the reference signal $r(k)$, the output signal $y(k)$, and the control input signal $u(k)$. Next, the fault alarm signal $a_m(k)$ is depicted. The fault detection signal $f_d(k)$ and the fault isolation signal $f_i(k)$ are also shown. The fault analysis signal $f_a(k)$ is not considered here. The fault occurs at time instant $t_k = 240$ s. The detection

delay is 1.5 s, and the isolation is achieved 8 s after fault occurrence. The fault is well isolated, since the fault isolation signal tends to the value 1 after the transient. All the fault detection and fault isolation signals are low pass filtered. The switching between the controllers (PI and LQGC) occurs at time instant t_k = 80 s, and this is evident in the variance of the control signal $u(k)$. This fault causes the saturation of the actuator.

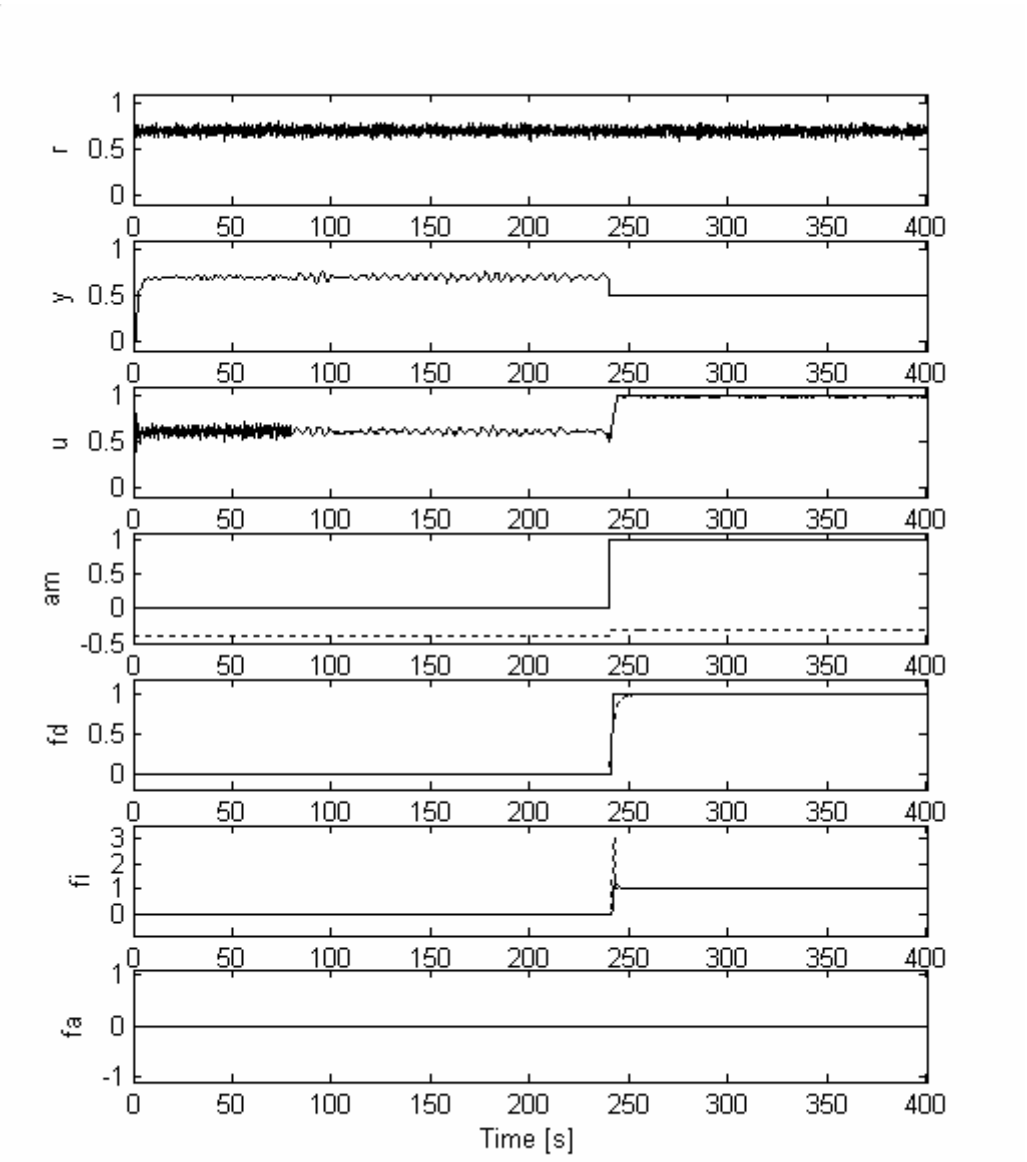


Fig. 4.14 - Input-output and FDD signals for fault F1.

In this experiment, in on-line operation, for each time instant k the input data vector is given by the vector of input-output signals [$y(k)$ $y(k\text{-}1)$ $y(k\text{-}2)$ $u(k\text{-}2)$]. The criterion used to select this vector is based on the data used in the ARX(2, 1, 2) model. Fig. 4.15 shows some signals for this experiment associated with the neural NLPCA. The first graph shows the input data **X** (green lines) as a function of time. The first graph also shows the predicted data values $\mathbf{X}_e$

(blue lines) by the neural network that implements the neural NLPCA approach. The square of prediction error (SPE, in blue line) then appears, given by $q(k)$ obtained from the NLPCA residual, and also the threshold values (red lines). In the third graph, the scores of the two principal components are shown as a function of time; PC1 in red line, and PC2 in green line. Finally, the last graph shows the two dimensional scores space, $\mathbf{T}_a(k) = [\mathbf{t}_1(k)\ \mathbf{t}_2(k)]$, where it can be observed the first principal component versus the second principal component. All the graphs show the signals for a time instant greater than 40 s. In nominal operation (fault- free case, F_0), the on-line scores are located around the cluster F0. When fault F_1 occurs, the on-line scores move towards the cluster F1, passing near the cluster F3 and this is the reason why in Fig. 4.14 the non filtered $f_{i0}(k)$ signal temporarily assumes the value 3.

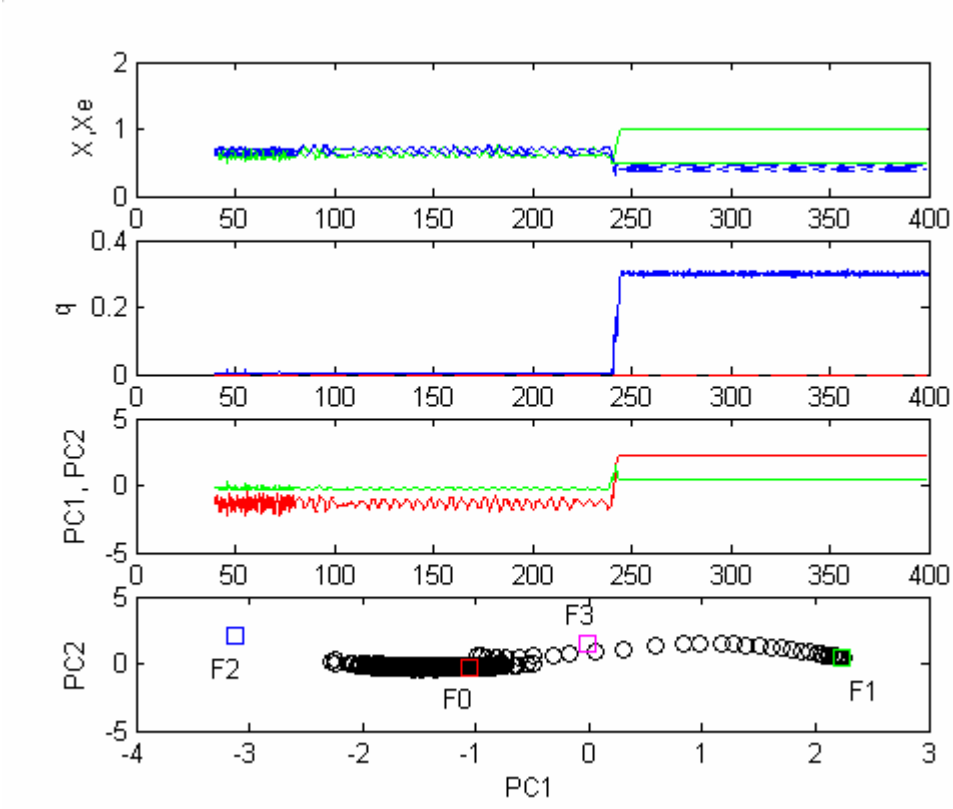


Fig. 4.15 - NLPCA scores and SPE for fault F1.

This FDD approach based on NLPCA applied to input-output data is not appropriate for detecting small parametric faults like changes on the filter bandwidth, etc. In fact, this type of fault must be detected using features based on model parameters, or other related variables.

Many authors argue that the parity equations and observers are most suitable for detection of abrupt additive faults, and the parameter identification is more appropriate for detecting parametric faults (Isermann, 1997; Gertler, 1998; Chen & Patton, 1999; Frank, et al., 2000a). In fact, this statement has been confirmed in this work.

Some tests have been performed in order to evaluate if this FDD methodology based on neural classical NLPCA and neural NNLDA is also able to detect some types of additive faults. In fact, the results show that this approach can detect and diagnose also some types of additive faults.

Results for an additive fault on the real actuator signal (termed F_4) corresponding to a voltage offset of -1.5 V are shown next. For a set point around 0.7 (nominal region) the real actuator voltage is around 3.2 V. When this fault occurs, the controller increase the control action in order to compensate the offset caused by the fault. Using the NLPCA model considered in Example 9, Fig. 4.16 shows the location of the scores associated with fault F_4 in the last graph (PC1 and PC2). It can be observed that the cluster is outside the nominal operation (cluster F0), and consequently this fault can be detected. The isolation can also be performed, since the cluster F4 is not superimposed on others, although it is near the cluster F1.

This figure also shows the reference signal $r(k)$ (red line), the output signal $y(k)$ (blue line), and the input signal $u(k)$ (green line). This fault almost cause a saturation of the control signal $u(k)$. The ARX(2, 1, 2) model M_{yu} parameters (th-yu) are also shown; the parameters $\theta(k) = \{a_1(k)\ a_2(k)\ b_2(k)\}$ are depicted using different colours, respectively, blue, green and red. The static gain (sg) is a good feature for FDD purposes in this case.

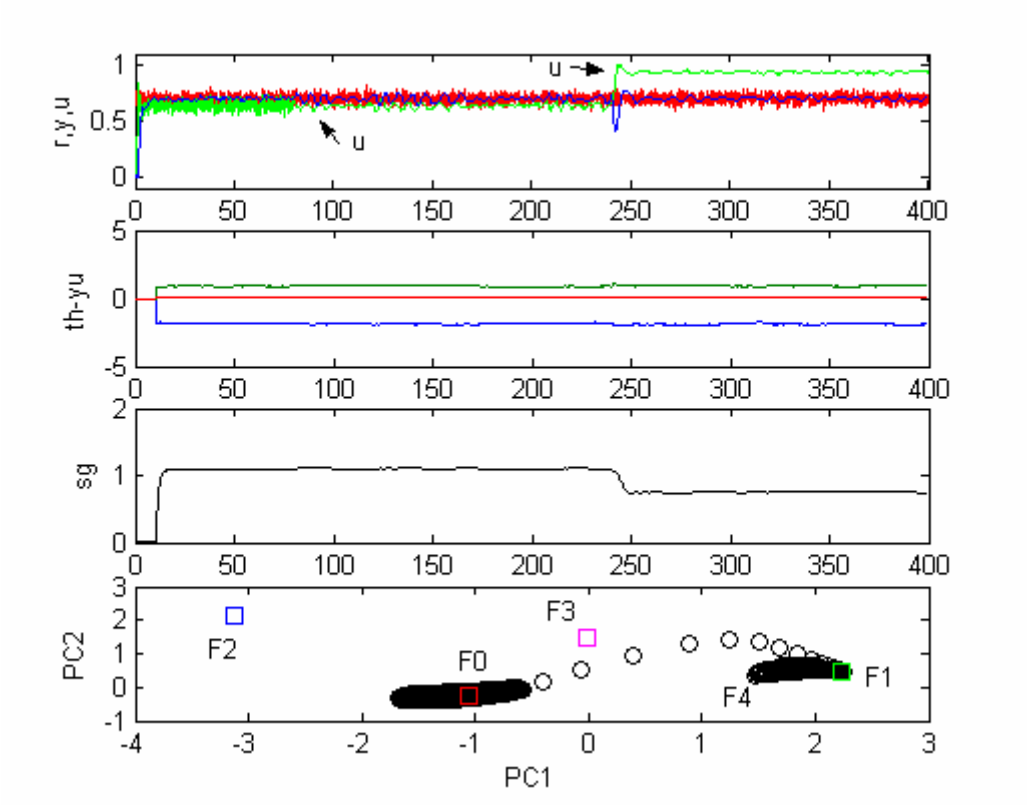


Fig. 4.16 - NLPCA scores and signals for fault F4.

4.5.5 FDD Approach Based on Neural NLPCA and Neural NNLDA

The classic approach to FDD based on NLPCA has been detailed in the section 4.5.4. The classic approach uses the input-output available data as input patterns.

Here, a new FDD approach based on neural nonlinear (NL) principal component analysis (PCA) and on neural nonlinear discriminant analysis (NNLDA) is proposed, applied to the ARX model parameters estimated on-line. In this approach, the input patterns are the parameters of ARX models.

The main contributions given here are: a) the use of the ARX model parameters $\theta(k)$ as features for FDD, instead of using the input-output available signals; b) the combination of the two neural methods NLPCA and NNLDA.

The problem to be solved here can be formulated as follows.

Problem 6. Given a nonlinear SISO continuous time system, without loss of generality, find a method for fault detection and diagnosis based on model parameters or related features, using nonlinear principal component analysis.

Assuming that the faults on the nonlinear continuous time system are reflected on the ARX model parameters estimated on-line, the following new fault detection and diagnosis approach is proposed here. This approach allows the detection of multiplicative (parametric) and also some kinds of additive faults.

The main idea is to detect and diagnose faults in the scores space and residual space of the NLPCA, assuming the features used for FDD purposes are the parameters of an ARX model. The neural nonlinear PCA approach allows a dimensionality reduction similar to the linear PCA case described in Chapter 3. This approach requires the on-line estimation of ARX parameters. Here, this is done by the sliding window PCR estimation algorithm. This new FDD approach was developed for nonlinear systems inspired by some ideas proposed in this dissertation for the linear systems described in Chapter 3, but now extended to deal with nonlinear systems.

The architecture proposed here to solve the problem under study is depicted in Fig. 4.17. Assuming a SISO continuous time system (plant), without loss of generality, with input signal $u(k)$ and output signal $y(k)$, the sliding window PCR estimation algorithm is used to estimate on-line the ARX model parameters, $\theta(k)$. A neural approach is used to implement the nonlinear principal component analysis (NLPCA) according to the Kramer's method (section 4.5.3). If the neural NLPCA method extracts two principal components from data, then a two-

dimensional scores space is obtained, $\mathbf{t}_a(k) = [t_1(k)\ t_2(k)]$, and one-dimensional residual space with SPE given by $q(k)$.

The features, neural NLPCA scores and SPE of the residual, define a three dimensional space, and this space is used for fault detection and diagnosis. The neural nonlinear discriminant analysis (NNLDA) previously described in section 3.3.2 is used for pattern classification, so to each on-line input feature vector $v(k) = [t_1(k)\ t_2(k)\ q(k)]$ a fault class F_i is assigned.

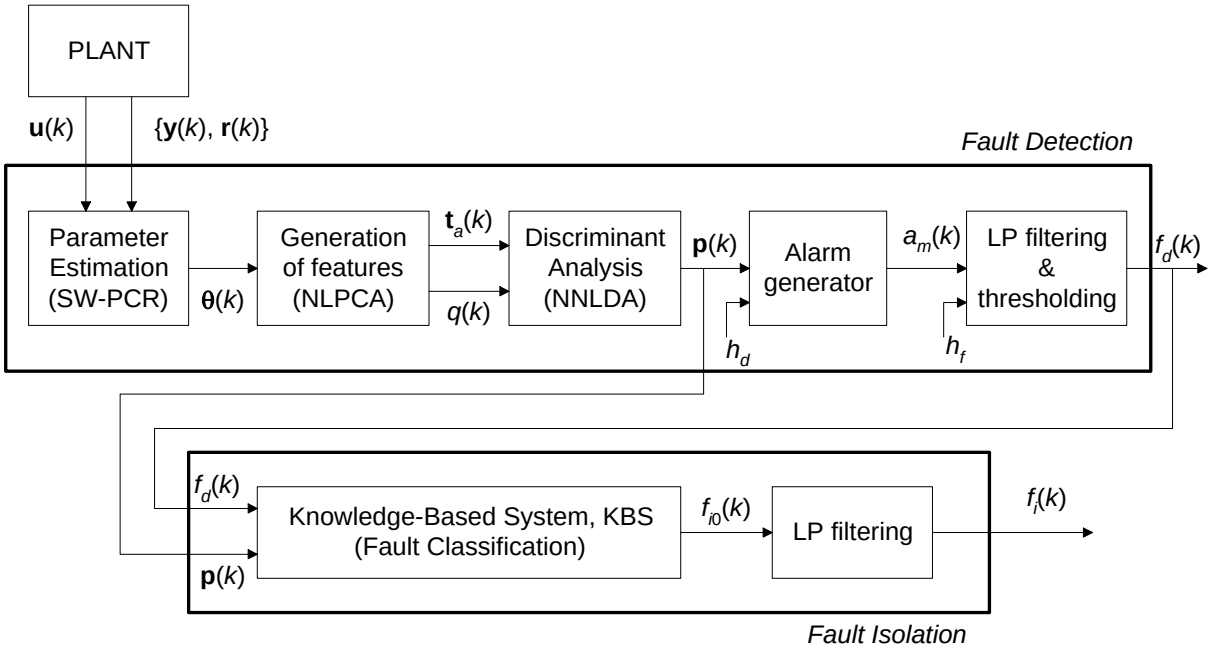


Fig. 4.17 - Architecture of FDD approach based on neural NLPCA and neural NNLDA.

The fault detection and diagnosis approaches are based on pattern classification via a neural nonlinear discriminant analysis (neural NNLDA) previous explained in section 3.3.2. A brief description is now given here.

Fault Detection Approach. The nominal behaviour corresponds to the fault F_0, and is characterized by a three dimensional cluster (pattern) in the features space defined by the NLPCA scores and the SPE of residual. The on-line output (fault class) vector of the neural nonlinear discriminant analysis model M_{NNLDA} is expressed in the form $\mathbf{p} = [p_1\ \ p_2\ \ p_3\ \ p_4]$ for the case of 4 faults, i.e., the i^{th} position of vector $\mathbf{p}$ is denominated $\mathbf{p}(\mathrm{i}) = p_i$. For the case of nominal operation, corresponding to fault F_0, $p_1 \approx 1$ and $p_i|_{i\neq 1} \approx 0$.

A fault alarm signal is generated if the deviation from the nominal behaviour exceeds a certain threshold (h_d), i.e., $p_1 < h_d$. Typical values of the threshold h_d are around 0.9. To obtain

a fault detection signal, the fault alarm signal is low pass filtered. The low pass filter $H_{lp}(z, \lambda)$ equations can be found in section 2.4.4. Finally, the low pass filtered signal is compared to a threshold (a typical value is 0.5) and the fault detection signal $f_d(k)$ is obtained.

Fault Diagnosis Approach. The task of fault isolation is executed after the task of fault detection. The isolation is based on a knowledge base system (KBS). The isolation is performed via the analysis of the fault class (pattern) generated by the neural nonlinear discriminant analysis model M_{NNLDA}. The isolation of fault number j is achieved if $round(p_{j+1}) = 1$ and $round(p_i)|_{i \neq j+1} = 0$, where $round(.)$ is the round function to nearest integer. The fault isolation signal is also low pass filtered. Only the fault isolation has been described and implemented, but this approach can be used also for fault analysis.
The new proposed FDD approach in the algorithmic form is described next, and obeys the architecture depicted in Fig. 4.17.

Algorithm 7. FDD neural approach based on neural nonlinear principal component analysis and neural nonlinear discriminant analysis (FDD-NLPCA-NNLDA).
In off-line operation, the following tasks must be executed:

a. For the nominal operation (fault F_0), train the neural auto-associative network $NN_{\{a\text{-}b\text{-}c\text{-}d\text{-}e\}}(\mathbf{W}_0, \ldots)$ for the nonlinear Kramer's PCA approach based on neural architecture depicted in Fig. 4.11, using the Levenberg-Marquardt optimization algorithm. This neural nonlinear PCA model is expressed here by M_{NLPCA}.
b. Proceed with the training of the neural network $NN_{\{a\text{-}b\text{-}c\text{-}d\}}(\mathbf{W}_d, \ldots)$ that implements the neural nonlinear discriminant analysis (NNLDA), using the clusters data for all the faults F_i of the set $F = \{F_0, F_1, \ldots, F_n\}$. The Levenberg-Marquardt optimization algorithm is used for the neural network training. This discriminant neural model is expressed here by M_{NNLDA}.
c. Determine the thresholds and the low pass filters parameters, in order to obtain a desired trade-off between rate of false alarms, rate of missed fault detections, and detection and isolation delays. Different sets of experimental nominal data must be used to compute and validate the thresholds and filters parameters.

Each time instant k, the following steps must be executed on-line:
1. Sample the process output signal $y(k)$.

2. Estimate the model parameters $\boldsymbol{\theta}(k)$ using the sliding window SW-PCR estimation algorithm based on input-output data, $u(k)$ and $y(k)$, or $r(k)$ and $y(k)$, depending on the type of ARX model.
3. Compute the two dimensional scores $\mathbf{t}_a(k) = [t_1(k)\ t_2(k)]$, and the one-dimensional SPE of residual, $q(k)$, using the nonlinear PCA model M_{NLPCA}.
4. Compute the class (pattern) output vector $\mathbf{p}(k)$ of the discriminant neural model M_{NNLDA}, for an input vector given by $v(k) = [t_1(k)\ t_2(k)\ q(k)]$.
5. Generate an alarm, $a_m(k) = 1$, if the first element of vector $\mathbf{p}(k)$ termed p_1 exceeds the threshold h_d , i.e., $p_1 < h_d$. A typical value for the threshold is 0.9.
6. Compute the fault detection signal by low pass filtering via $H_{lp}(z, \lambda)$ the fault alarm signal, and by thresholding. The thresholding is expressed by the rule: if $a_m(k) > h_f$ then $f_d(k) = 1$ else $f_d(k) = 0$. A typical value for the threshold is around 0.5.
7. If a fault is detected, i.e. $f_d(k) = 1$, then proceed to fault isolation. Using a knowledge based system (KBS) classify (map) each output vector pattern $\mathbf{p}(k)$ into the respective fault class F_i. The KBS system has been implemented using if-then rules, but fuzzy if-then rules or neural networks can also be used if a large number of faults must be isolated. The signal $f_{i0}(k)$ is then obtained. Finally the fault isolation signal $f_i(k)$ is obtained by low pass filtering the signal $f_{i0}(k)$.

■

An example of application of the proposed FDD methodology is given next, with application to a nonlinear real motor setup (DCM-RA setup).

Example 10. FDD approach based on neural nonlinear PCA and neural nonlinear discriminant analysis (NNLDA) applied to ARX model parameters.

In this example, some experiments performed with the dc motor (DCM-RA) setup used also in Example 9 are shown; the setup is described in more detail in section 5.3.3. All operating conditions are the same as used in Example 9. An LQG controller is also used to control the DC motor setup.

Typically, the classical NLPCA approach based on input-output data is not able to detect and isolate some types of parametric faults. In this example, it will be shown that the new FDD approach proposed, based on model parameters, can detect and isolate parametric faults, such as the changing of a time constant of a sensor.

Four faults are considered here in this example. The fault F_0 corresponds to the nominal operation, for a set point around 0.7. Faults F_1 and F_2 are cases of blocked sensors at values 0.5 and 0.8. The last fault, F_3, corresponds to the removal of the digital filter $H_{lp}(z, \lambda)$ from the process output, putting a zero value on the filter design parameter, $\lambda = 0$, and consequently turning the system faster.

The NLPCA approach is applied here to an ARX(2, 1, 2) model that relates the input signal $u(k)$ and the output signal $y(k)$. For the neural network used for NLPCA at the end of 200 epochs of training, a SSE error of 1.1×10^{-2} was achieved.

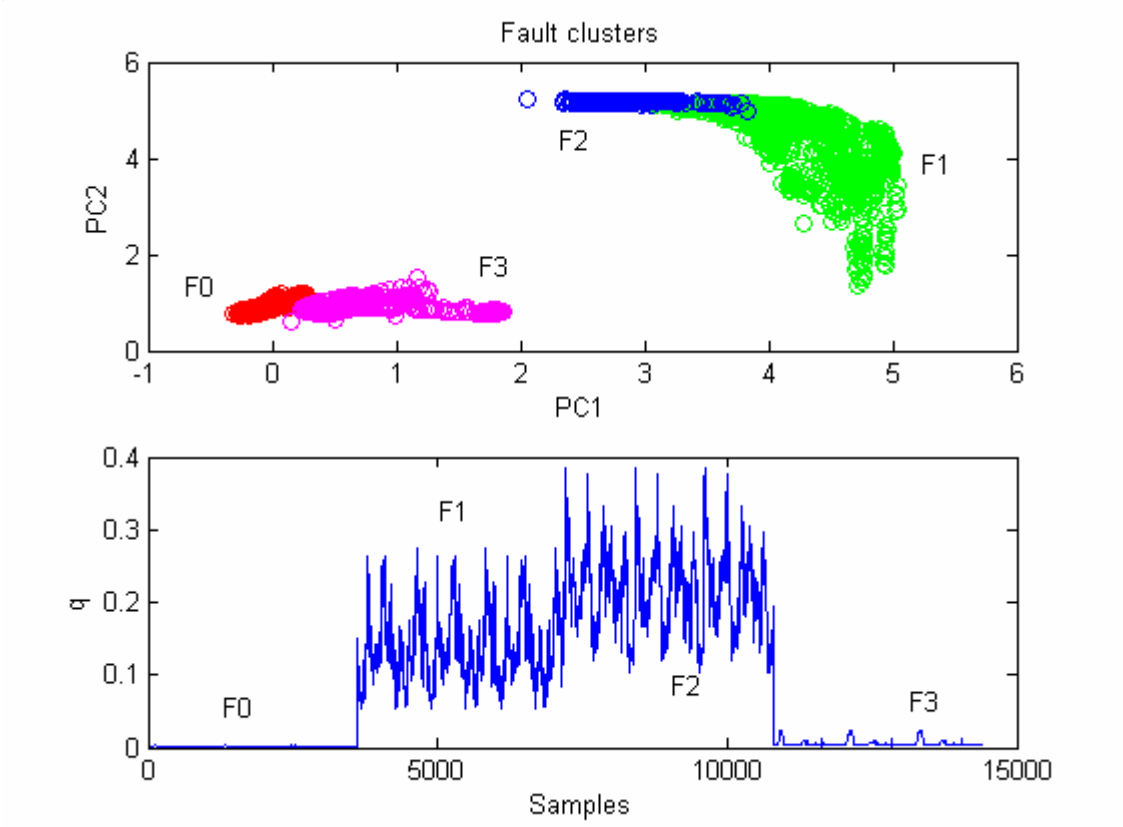


Fig. 4.18 - Fault clusters in scores space, and SPE signal.

Fig. 4.18 shows the scores (PC1 and PC2) and the SPE (q) signal used for training the neural classifier used of the neural NNLDA approach using pattern classification. These data have been captured during different faulty situations. The clusters associated with faults F_0 and F_3 are not overlapped, but a small overlapping exists between fault clusters F1 and F2. When the clusters are very near, or slight overlapping occurs, large sum of squared errors (SSE) are obtained in the neural network classifier training. Here, for 500 epochs a high SSE error of 11.6 was obtained, for 14400 (3600×4) samples. For another training, a mean squared error

(MSE) of 1.99×10^{-4} was achieved for the same training data. A high SSE error in the neural network error means that the neural classifier cannot separate some patterns very well, causing fault isolability problems. In this experiment, the cluster superposition occurs because these faults cause saturations and, consequently, a degradation of the persistent excitation conditions. For weak persistent excitation conditions, an increase in the variance of the ARX model parameters occurs, and that is reflected on the NLPCA scores and SPE signal.

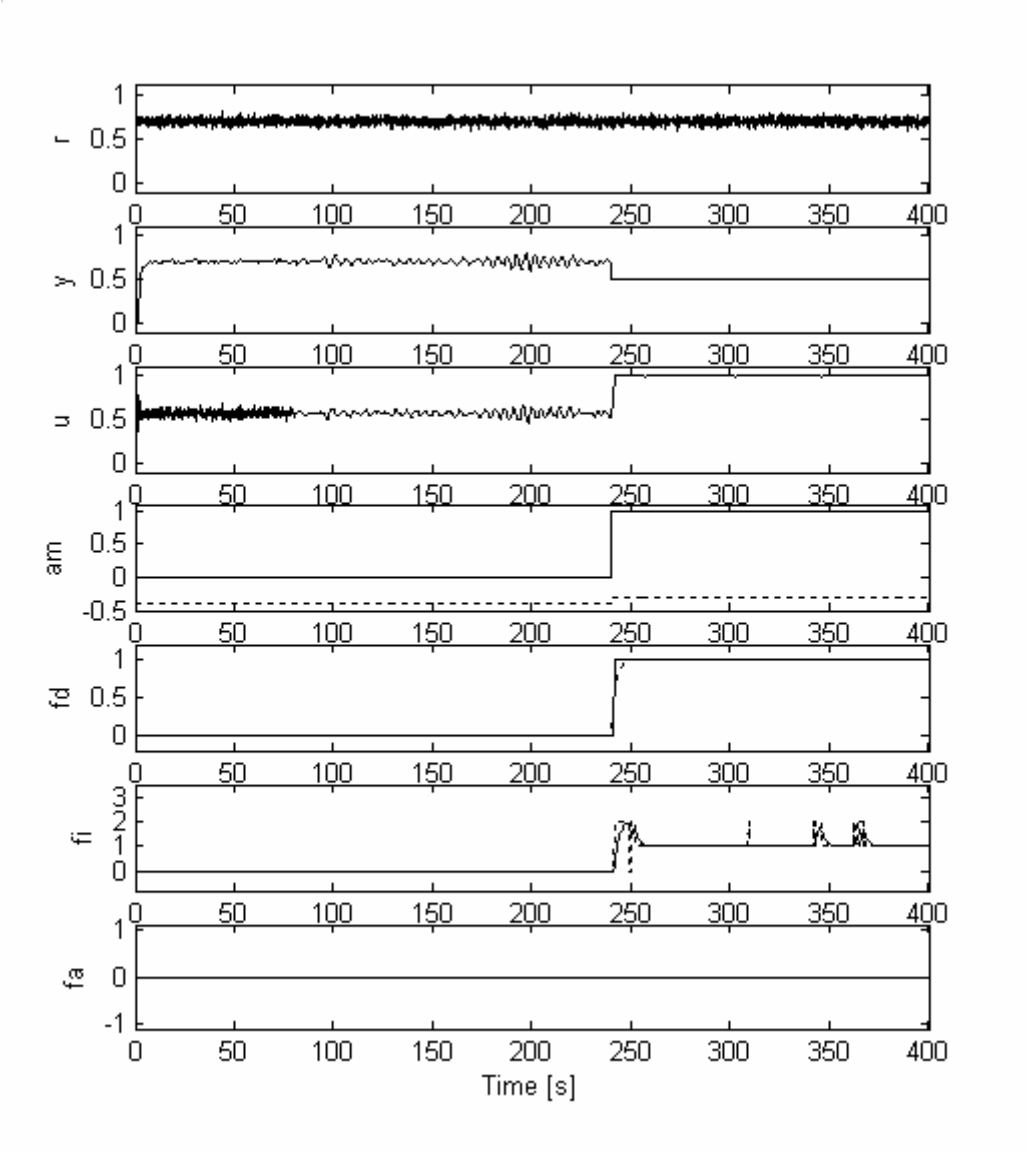


Fig. 4.19 - Input-output and FDD signals for fault F1.

Results obtained with the fault F_1 are depicted in Fig. 4.19, where a sensor stays blocked at the value 0.5. The graph show, respectively, the reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. Next, the fault alarm signal $a_m(k)$, the fault detection signal $f_d(k)$, and the fault isolation signal $f_i(k)$ are shown. The fault analysis $f_a(k)$ is not considered here.

The fault occurs at time instant $t_k = 240$ s, the detection delay is 1.5 s, and the proper isolation occurs 15 s after fault occurrence. Here the great advantage of low pass filtering the fault isolation signal is clear. Some wrong fault isolations occured during the experiment, and the low pass filtering minimizes this problem. This isolation problem is to be expected since the fault clusters associated with faults F_1 and F_2 are very near, as depicted in Fig. 4.18.

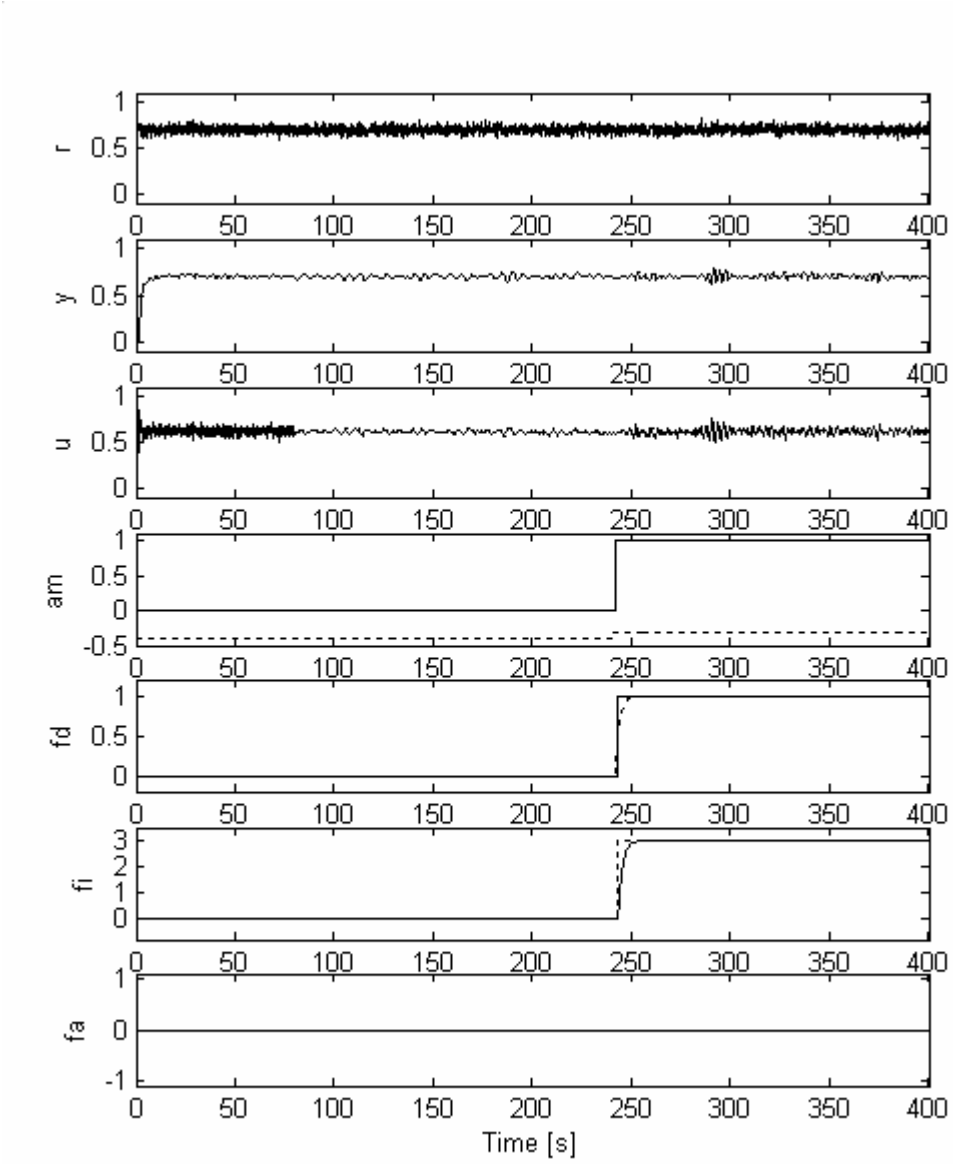


Fig. 4.20 - Input-output and FDD signals for fault F3.

In Fig. 4.20 results obtained with the fault F_3 are shown. In this fault the system bandwidth is increased by removing the low pass digital filter $H_{lp}(z, \lambda)$ from the process output. The classical NLPCA approach based on input-output data was not able to detect this fault, but the new proposed approach can detect and isolate this fault well, since the ARX model parameters are sensitive to the change of the system bandwidth, as it will be shown next. The

signals presented in Fig. 4.20 are the same as those described in the experiment shown for fault F_1. The detection delay is 3.5 s, and the isolation takes place 12 s after fault occurrence.

Fig. 4.21 shows in the first graph the ARX model parameters that are the input data (green lines) for the NLPCA approach. The predicted values (blue lines) are also shown. Next the SPE $q(k)$ (blue line) appears and also the thresholds (red lines). The third graph show the scores as a function of time (PC1 in red line, and PC2 in green line). The last graph shows the scores space (PC1 and PC2). In fact, the clusters for faults F_0 and F_3 are very near, although they are not superimposed. The centres for each fault cluster are also depicted.

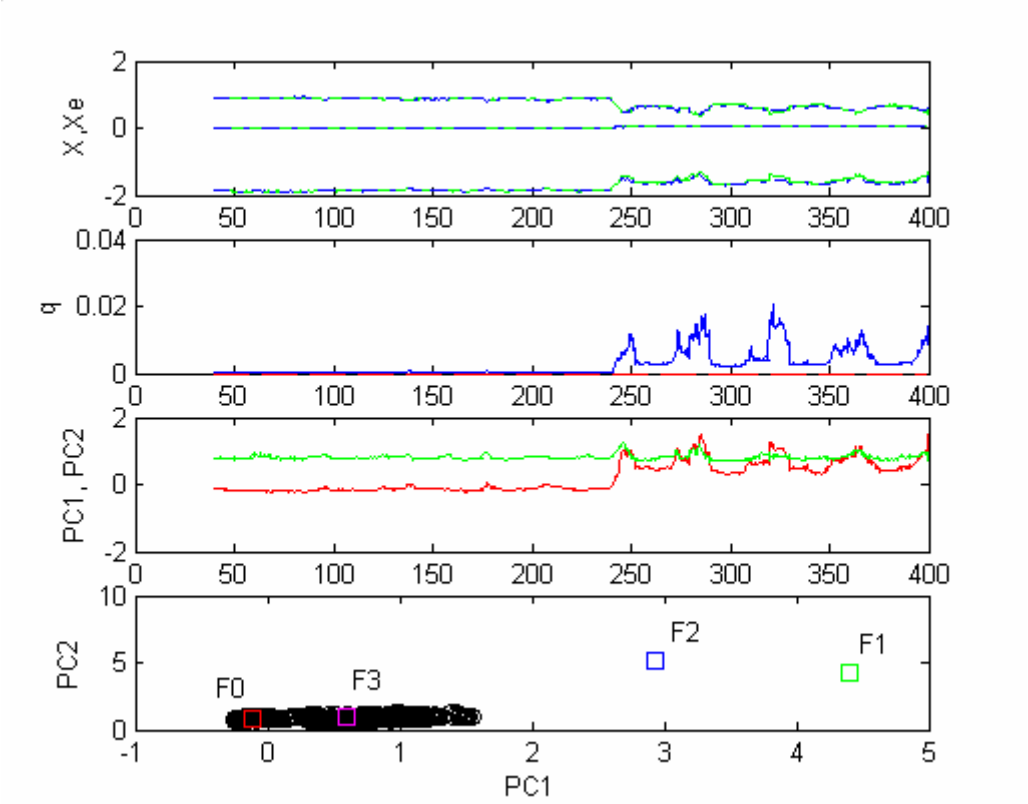


Fig. 4.21 - NLPCA signals for fault F3.

The FDD methods based on parameter identification techniques are most suitable to detect parametric faults, and some structural faults, that cause deviations on the model parameters. But these methods can also be used for detection and diagnosis of additive faults. The great advantage of identification methods is that they can be used to detect both multiplicative and additive faults (Gertler, 1998).

Some tests have been performed in order to evaluate if this FDD methodology based on neural NLPCA and neural NNLDA is also able to detect some types of additive faults on sensors and actuators. The experiments show that this is possible in some faulty situations, but

not possible in others. Additive faults that cause significant variations of the ARX model parameters can be detected and diagnosed using this FDD approach, for example in plants with strong nonlinear behaviour.

Faults that cause small parameter deviations cannot be detected and diagnosed very well, as shown in the next experiment. Fig. 4.22 shows results obtained with the FDD approach proposed here. The same additive fault considered in Example 9 (section 4.5.4), a voltage offset of -1.5 V on the real actuator signal, is tested here. In this figure, the reference signal $r(k)$ (red line), the output signal $y(k)$ (blue line), and the input signal $u(k)$ (green line) are observed. This fault almost causes a saturation of the control signal $u(k)$. The ARX model M_{yu} parameters (th-yu) are also shown, as well the corresponding static gain. The static gain is a good feature for FDD purposes in this case. The last graph shows a zoom view of the scores space, where the superposition between clusters F0 and F4 can be observed; the red square symbol indicates the cluster F0 centre. Hence, this fault cannot be correctly detected and isolated using this FDD approach. FDD approaches based on parity equations, observers or on the NLPCA approach based on input-output data, usually, exhibit a better performance for this type of additive faults.

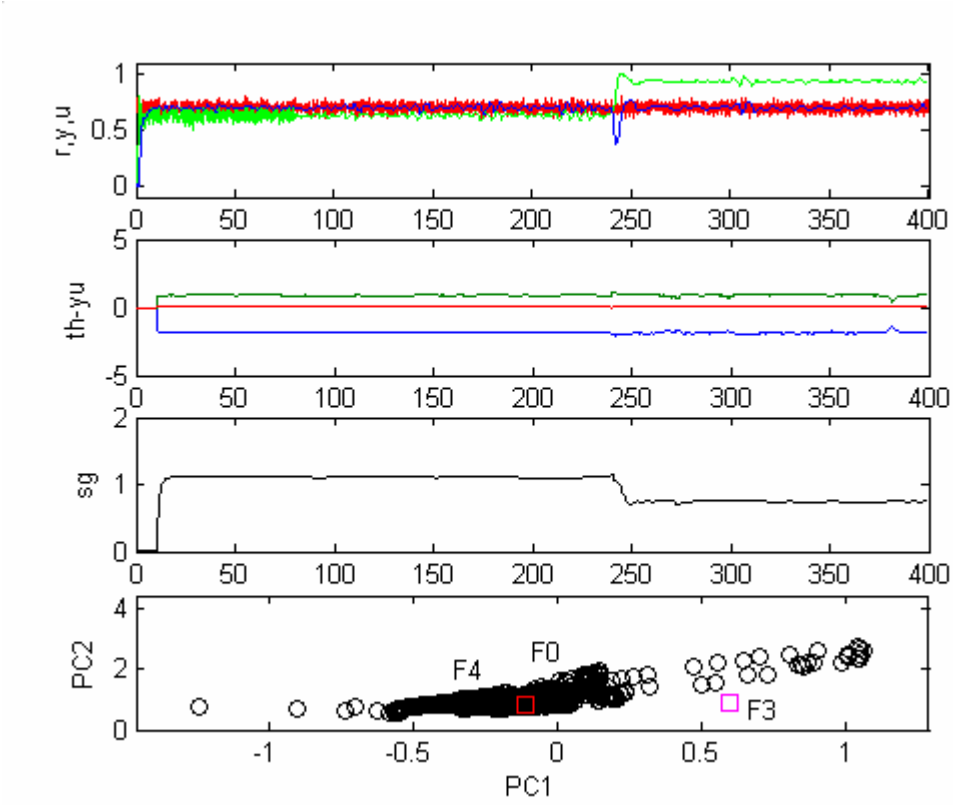


Fig. 4.22 - NLPCA scores and signals for fault F4.

4.6 Conclusions

In general, most of the fault detection and diagnosis (FDD) approaches for linear systems can also be applied to nonlinear systems assuming they work on steady-state operation. If it is possible, nonlinear FDD approaches must be used when dealing with nonlinear systems. Most of the research efforts have been made in linear FDD methodologies applied to linear systems, and little research has been oriented to the development of nonlinear FDD approaches.

The main contributions to the FDD nonlinear techniques given here are based on nonlinear NARX neural models. The neural networks prove to be an effective tool to perform the tasks of fault detection and fault diagnosis. The great advantages of neural networks are their vast potential for nonlinear modeling, and their robustness properties in terms of noise influence and non modeled dynamics. Neural networks have been used for output prediction, for nonlinear principal components analysis (NLPCA) and for nonlinear discriminant analysis (NNLDA).

A Neural Recurrent Output Predictor (NROP) has been proposed in this work for prediction of output signals in SISO or MISO systems. The NROP predictor proposed contains an embedded parallel neural model and external feedback. The NROP gain is a design parameter guaranteeing the stability and adjustment of the convergence properties.

A new FDD approach for nonlinear systems based on a bank of Neural Recurrent Output Predictors (NROPs) was proposed. This FDD approach, based on input-output data, is appropriate for additive faults, for some multiplicative faults, and for some faulty cases where other approaches normally fail, like situations of saturations, structural faults and oscillatory behaviours. The great potential of this FDD approach based on NROPs is the fact that each neural NROP predictor contains an embedded neural model that is able to capture nonlinear dynamic behaviours.

Inspired by the fault detection approach based on PCA applied to the ARX parameters proposed in section 3.5 for linear systems, this chapter describes a new combined FDD approach for nonlinear systems based on neural nonlinear PCA (NLPCA) applied to ARX parameters, and on neural nonlinear discriminant analysis (NNLDA). In order to obtain a reasonable performance for this FDD method, the identification algorithm must give parameter estimates with small variances, and the patterns for each fault must be well separated in the features space. This approach is more suitable to detect and diagnose parametric faults, in spite of being able to deal also with some types of additive faults, especially those whose symptoms are deviations on the ARX model parameters.

5 Experimental Results

The experiments help to understand the theory better. (L. B. Palma).

5.1 Introduction

Typically, most industrial dynamic systems work under closed-loop control, as depicted in Fig. 5.1. This is the main reason why the focus on this dissertation is on the development of fault detection and diagnosis (FDD) methods operating in closed-loop, on-line and with real-time constraints.

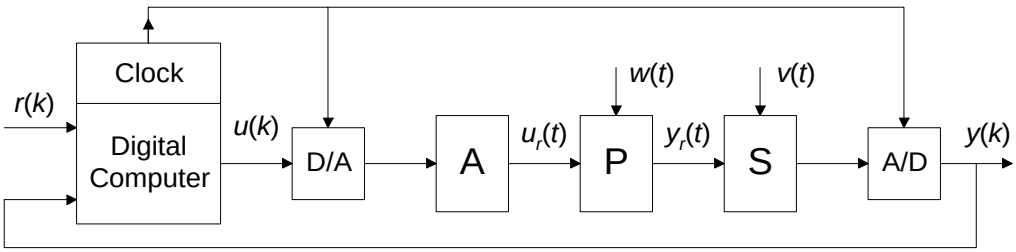


Fig. 5.1 - Closed-loop control architecture.

Assuming a SISO system, without loss of generality, the process plant is represented by the block P, and the blocks A and S represent, respectively, the actuator and the sensor. The blocks A/D and D/A are the analog-to-digital and digital-to-analog converters. The digital computer implements the supervision, control and FDD algorithms. The digital signals are also represented in the figure: the reference signal, $r(k)$, and the input and output signals, $u(k)$ and $y(k)$, respectively. The other signals are analog signals: the real process input, $u_r(t)$, and the real process output, $y_r(t)$. The signals, $w(t)$ and $v(t)$, are, respectively, the disturbance input to the plant, and the disturbance or noise in the sensor. Usually, the signals $u_r(t)$, $y_r(t)$, $w(t)$ and $v(t)$ are not available for FDD purposes. The typical situation is to perform the FDD tasks based only on the available input-output discrete time signals $u(k)$ and $y(k)$, or based on the signals $r(k)$ and $y(k)$.

In this work, a white (Gaussian) noise with normal distribution $N(\mu = 0, \sigma^2)$ (zero mean and variance σ^2) has been used as a dither signal added to the reference signal $r(k)$. This dither is

used to guarantee the good persistent excitation conditions needed to perform identification in closed-loop operation. A white noise was also added to the sensor signal for the plant models.

In all the experiments performed in this work, the saturation does not occur exactly at the values zero (0) and one (1). The saturation occurs in positive values, around zero and around one. A dither signal has been added to the saturation limits zero (0) and one (1), in order to guarantee that under saturation situations the persistent excitation condition is reasonable. The variance of this dither signal is 1×10^{-5}. The persistent excitation conditions must be satisfied in parameter estimation algorithms for fault detection and diagnosis, and also for adaptive control.

In this work, the proposed FDD methodologies have been first tested on model plants, before the test on real plants. This strategy allows a better validation of the methods and also saves a lot of time. Compared to real plants, the models have a great advantage: a greater number of fault types can be tested.

Most of the experiments shown in this dissertation have a duration of 400 s. A duration of 600 s is used for the three-tank benchmark experiments. For most cases, the fault occurs at around 60% of the duration, and the fault detection and diagnosis task is activated from 40% of the duration.

All the FDD approaches proposed require the low pass filtering of signals to obtain the fault detection and the fault isolation signals. This is done using a digital low pass filter, $H_{lp}(z, \lambda)$, and the design parameter (pole location) used is around $\lambda = 0.9$. The filter equations can be found in section 2.4.4.

The sampling period used in all experiments is $T_s = 0.11$ s.

5.2 Matlab Programming and Hardware Interfaces

The algorithms of modelling, simulation, control, fault detection and diagnosis and supervision developed in this work have been implemented in the Matlab® environment, in discrete time and with real time constraints.

One of the aims of this work was the implementation of a software framework able to emulate real situations. So, the process models have been implemented in continuous time, and simulated using the Matlab function “ode45(.)”. The function “ode45(.)” solves ordinary differential equations.

The interface to the real setups has been done via a data acquisition board PMD-1208LS connected to the USB port, manufactured by Measurement Computing®.

The Matlab code has been developed in a modular way using scripts, functions, and structs. More than 24000 lines of code have been written throughout the work. Structs are very important in the programming since they support a set of variables (or other structs) within. As an example, a "sensor struct" can be constructed according to: sensor.id = 'sensor.id'; sensor.y(time.k) = 0.8.

The time consumed by the routines in each program cycle depends on the granularity of the called routines, but is less than 0.1 s, for a PC equipped with a Pentium processor running with a clock of 2.8 GHz and 1 GB of RAM. The sampling period used in all experiments is $T_s = 0.11$ s.

5.3 Models and Real Setups

In this work system models and real plants were used to test the new proposed FDD methodologies. It was assumed that we are dealing with continuous time plants and models. From the point of view of FDD algorithms they were implemented in discrete time using mainly linear ARX models and neural nonlinear NARX models.

For the case of linear ARX models, the choice of the model orders was based on good low order models suggested by the system identification toolbox of Matlab. This toolbox tests a set of models with different orders and delays, and indicates the best models. In this set, three models usually satisfy the criteria: the model with the Best Fit, the model that minimizes Akaike's criterion (AIC), and the model that minimizes Rissanen's MDL criterion. The model selected and used in the FDD approaches proposed here is one of the low orders in the set.

Next the models and real setups used in the experiments are presented.

5.3.1 A Continuous-Time First-Order Model

A simple continuous-time LTI SISO model for a first-order system can be expressed by a transfer function in the frequency domain s, given by Eq. 5.1 where K is the static gain and τ is the time constant.

$$G_0(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \qquad \text{Eq. 5.1}$$

Although it is a simple model, it has been used in this work to illustrate various concepts. A discretization by the zero-order-hold (ZOH) method will give an ARX(n_a=1, n_b=1, n_d=1) model (Eq. 5.2), with the static gain given by $G_0(z=1) = b_1 / (1 + a_1) = K$.

$$y(k) = -a_1\, y(k\text{-}1) + b_1\, u(k\text{-}1) + e(k) \qquad \text{Eq. 5.2}$$

If the nominal parameters assume the values $K = 1$ and $\tau = 1$ s then the ARX(1, 1, 1) parameters are given by $[a_1\ b_1] = [-8.96\times10^{-1}\ 1.04\times10^{-1}]$, for a sampling period of $T_s = 0.11$ s. For some experiments in this work, other high order discrete models are used for modeling this first order continuous time model.

5.3.2 A Continuous-Time Linear DC Motor Model

Mechanical and electrical processes, especially the DC motor, have been used in many simulations and real applications related to FDD & FTC, as described in several publications. Here a simplified model of a DC motor in continuous-time is presented. In armature controlled DC motors, the applied voltage $u_a(t)$ controls the angular velocity $\omega_r(t)$ of the shaft. A simplified continuous-time transfer function of the DC motor is given by a 2^{nd} order system (Kuo, 1995):

$$G_m(s) = \frac{\omega_r(s)}{u_a(s)} = \frac{K_m}{(L\,J)\,s^2 + (L\,K_f + R\,J)\,s + (R\,K_f + K_m\,K_b)} \qquad \text{Eq. 5.3}$$

where K_m is the torque constant, L is the armature inductance, R is the armature resistance, J is the rotor inertia, K_f is the viscous-friction coefficient, and K_b is the back-emf constant. The nominal values, in S.I. units, used for the simulations are: $K_m = K_b = 0.1$ N m / A, $L = 0.5$ H, $R = 2\ \Omega$, $J = 0.02$ kg m^2 / s^2, and $K_f = 0.2$ N m s. The poles are located at $s_1 = -9.8$, $s_2 = -4.1$. A gain with value $K_s = 4.1$ has been added in series at the output to guarantee a unitary static gain.

In the simulations presented, at the model input and at the model output of the DC motor model, digital low pass filters $H_{lp}(z, \lambda)$ have been added in series in order to guarantee a slower dynamics. The design parameter λ determines the pole location. For a value $\lambda = 0.7$, converting the filter to continuous time, via the zero-order-hold method, a transfer function

$F_f(s)$ with a pole located at $s_f = -3.24$ it is obtained, for a sampling period of $T_s = 0.11$ s. The general architecture used for the simulations with the DC motor model is depicted in Fig. 5.2. The blocks A/D and D/A represent the analog-to-digital converter and the digital-to-analog converter, respectively.

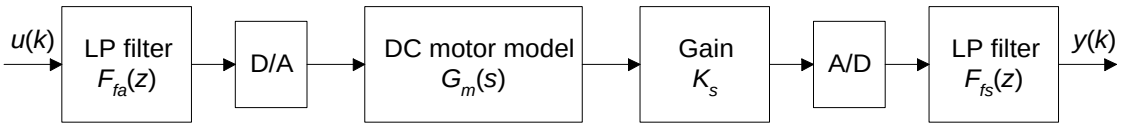
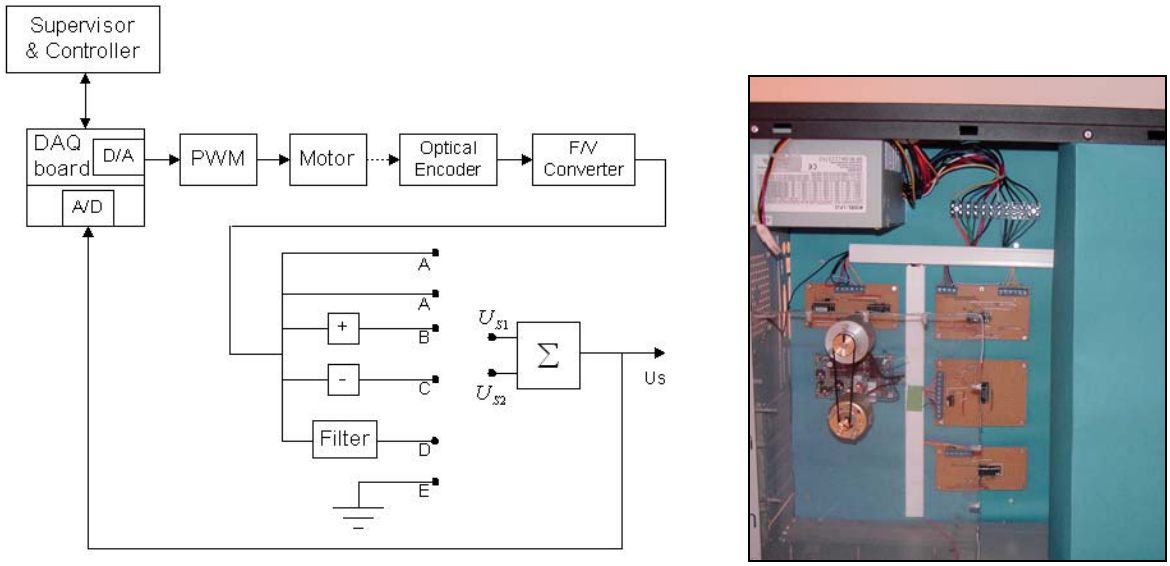


Fig. 5.2 - Architecture of the DC motor model with LP filters.

5.3.3 A Nonlinear DC Motor Setup

A DC motor setup, with the short-name DCM-RA, was developed in the control laboratory of DEE-FCT-UNL, during the years 2004-2005, by Rui Almeida within the scope of the final project course. Fig. 5.3 depicts the overall architecture of the system and a lateral image of the setup. The main goal was the development of real equipment for testing FDD algorithms, for both additive and parametric faults.



a) b)

Fig. 5.3 - DCM-RA motor setup: a) architecture; b) lateral image.

The DCM-RA setup is a nonlinear system, since the static gain depends on the operating point. Some experiments have shown that for the vector of scaled set-points

$\mathbf{r}$ = [0.4 0.5 0.6 0.7 0.8 0.9] the vector of static gains is approximately given by $\mathbf{s}_g$ = [0.81 0.93 1.04 1.15 1.21 1.27] . It also presents a time-varying behaviour depending on the operating temperature. These behaviours are due to the electronic and mechanical components. A temperature increase causes an increase on the static gain.

The supervisor and the controller are implemented in Matlab® software on a PC. The interface is made via a data acquisition usb-board (PMD-1208LS from Measurement Computing®). The actuator is a power driver based on a pulse-width modulation (PWM) circuit. The motor is a DC motor with permanent magnets supporting input voltages between zero and twelve Volt. The speed sensor includes an optical encoder and a frequency-to-voltage converter circuit. The frequency-to-voltage converter has a slow dynamic.
In order to convert the signals to a scaled range [0; 1], it was necessary to use gains in software for the actuator and the sensor. The actuator gain is 5 since the control voltages are in the range [0; 5] V, and the sensor gain is 0.1 since the sensor voltages are in the range [0; 10] V. It is assumed in this work that the nominal operating region, for the set-point (reference) signal, of this DC motor setup is in the scaled range from 0.5 to 1.0. A reference (set-point) value of 0.4 is very near the critical minimum admissible speed. For lower set-points the system cannot work because the applied voltage is not sufficient to guarantee the motor rotation.

Additive faults and multiplicative faults in hardware can be tested using the DCM-RA setup. A switch allows a change among three available analog anti-aliasing filters at the process output. The filter cutoff frequencies belong to the set $\{f_{c1}, f_{c2}, f_{c3}\}$ = {1, 12, 30} Hz. For a sampling period of T_s = 0.11 s, the most appropriate anti-aliasing filter is the filter with cutoff frequency f_{c1}, since the Nyquist-Shannon theorem theoretically imposes the condition to avoid aliasing: $F_s > 2 \times F_{max}$, i.e., the sampling frequency F_s must be, at least, twice the maximum signal frequency F_{max}. The anti-aliasing filter with cutoff frequency f_{c1} was used in the experiments carried out in this work. In practice, the relation F_s / F_{max} must be greater than 10 or more, depending on the applications. For T_s = 0.11 s, F_s = 9.1 Hz, and consequently the cutoff frequency f_c of the anti-aliasing filter must obey the relation $f_c < F_s / 2 \Leftrightarrow f_c < 4.5$ Hz. In the experiments, a digital first order low pass filter $H_{lp}(z, \lambda)$, with pole located at p_z = 0.7, was added in series, in software, to the system output.

5.3.4 The Three-Tank Benchmark

The Three-Tank Benchmark has been elaborated within the COSY program of the European Science Foundation, in order to perform initial investigations of the control reconfiguration problem under severe structural faults. The general problem to be solved is to find a new control strategy if a fault in the technical plant has occurred.

The benchmark problem is made up of the three coupled tanks depicted in Fig. 5.4, being a multi-input multi-output (MIMO) system. A more detailed description of the system can be found in the paper written by Heiming & Lunze (1999). These tanks are connected by pipes which can be controlled by several on-off valves. Water can be let into the left and right tank using two identical pumps (P_1 and P_2). Measurements available from the process are the continuous water levels $\{h_1(t), h_2(t), h_3(t)\}$ in each tank, and discrete water levels h_d assuming values $\{low, medium, high\}$ from two capacitive proximity switches attached to the central tank. For the central tank T_3, these qualitative values are *low* = [0; 9] cm, *medium* = [9; 11] cm and *high* = [11; 60] cm. The aim is to provide a continuous water flow Q_N to a consumer by maintaining the level $h_3(t) = medium$ in tank T_3. The reservoir-tank T_1 is filled by pump P_1 up to a nominal water level of h_1 = 50 cm.

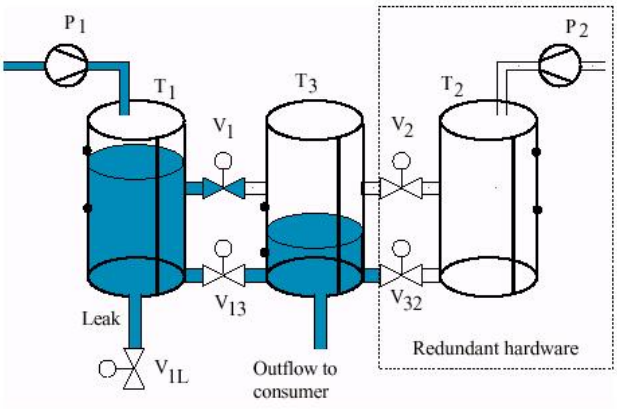


Fig. 5.4 - Three-tank benchmark system.

In the fault-free situation, only the left tank T_1 and the middle tank T_3 are used as shown in Fig. 5.4; tank T_2 and pump P_2 are not used. A continuous time PI-controller or other type of controller can be used to control the level around 0.5 m at tank T_1. A switching (on-off) controller opens and closes valve V_1 thus maintaining the level around 0.1 m at the central tank T_3. All other valves are closed, and the right tank T_2 is empty. Tank T_2 and pump P_2 are used as redundant hardware, if needed. The connecting pipes between the tanks are placed at the bottom of the tanks and at a height of 30 cm (at the middle of the tanks).

Three typical faults scenarios are considered by Heiming & Lunze (1999): blocking valves and leaks. The first fault scenario F_1 is valve V_1 closed and blocked, and the aim is to maintain the water level in tank T_3 still *medium*. The fault F_2 is valve V_1 opened and blocked, and the aim is to maintain the water level in tank T_3 still *medium* (around 0.1 m). The last fault scenario, fault F_3, is a leak in tank T_1, and the aim is to maintain the water level in tank T_3 still *medium* and guarantee a minimal loss of water from tank T_1.

The fault tolerance for this type of faults in this benchmark requires system reconfiguration. The reconfiguration task involves finding a new control structure by the selection of actuators and sensors, new control laws and new set-points for the control loops in such a way that the control aims are reached. The idea of reconfiguration cannot be satisfied by simply changing the parameters of the controller, but a structural change of the system is necessary.

From the theoretical point of view, the three coupled tanks are a typical hybrid system (Heiming & Lunze, 1999). Depending on the water levels and the position of the valves, different nonlinear state space models are valid. In general, the water flow Q_{ij} from tank T_i into tank T_j can be calculated using the Toricelli law

$$Q_{ij} = a_z \, S \, sgn(h_i - h_j) \sqrt{2 \, g \, |h_i - h_j|} \, , \qquad \text{Eq. 5.4}$$

where a_z is a flow correction term, S the cross-section area of the connecting valve, g the gravity constant, h_i and h_j the water levels above the connecting pipe, and *sgn*(.) the sign function. The change of water volume V in a tank can be computed as

$$\frac{dV}{dt} = A \frac{dh}{dt} = \sum Q_{in} - \sum Q_{out} \, , \qquad \text{Eq. 5.5}$$

where $\sum Q_{in}$ is the sum over all water inflows into the tank and $\sum Q_{out}$ the sum over all water outflows from the tank. The water level in the tank is h and A is the cross-section area of the cylindrical tank.

In general, the nonlinear differential equations for the three tanks are

$$\frac{dh_1(t)}{dt} = \frac{{Q_1}^{P_1} - {Q_{13}}^{V_1} - {Q_{13}}^{V_{13}} - {Q_{Leak}}^{V_{1L}}}{A}$$

$$\frac{dh_2(t)}{dt} = \frac{{Q_2}^{P_2} - {Q_{23}}^{V_2} - {Q_{23}}^{V_{32}}}{A} \qquad \text{Eq. 5.6}$$

$$\frac{dh_3(t)}{dt} = \frac{{Q_{13}}^{V_1} + {Q_{13}}^{V_{13}} + {Q_{23}}^{V_2} + {Q_{23}}^{V_{32}} - Q_N}{A} \,,$$

where the flows Q_{ij} depend on the levels h_1, h_2 and h_3 in the tanks, as well as on the position of the valves. The valve positions $Pos(.)$ can be 0 (closed) or 1 (opened), and the outflow from pumps P_1 and P_2 are the terms ${Q_1}^{P_1}$ and ${Q_2}^{P_2}$. Depending on the water level $h_i < 30$ cm or $h_i \geq 30$ cm, there are eight different operating modes for the three-tank system (Heiming & Lunze, 1999).

The nominal operating mode is given by the equations

$$h_1 \geq 0.3 \text{ m}, h_2 < 0.3 \text{ m}, h_3 < 0.3 \text{ m}$$

$$Q_{13}{}^{V_1} = a_z S \, sgn(h_1 - 0.3) \sqrt{2 g |h_1 - 0.3|} \, Pos(V_1) \qquad \text{Eq. 5.7}$$

$$Q_{23}{}^{V_2} = 0$$

where $Pos(V_1)$ denotes the position of valve V_1. For this nominal operating mode, the level in tank T_1 must be around 0.5 m, and the level of tank T_3 must be medium (i.e., in the range [9;11] cm).

The model of the three-tank benchmark available on the internet has been implemented using the Simulink® software. In this work, a Matlab version was developed based on the function "ode45(.)". The specifications used for the simulations are shown in Tab. 5.1.

In the simulations presented in this dissertation, all the values have been scaled to the range [0; 1]. Let us again remember, that a continuous-time model is used for the three-tank system in the experiments, but all the supervision (including FDD and FTC tasks) and the control algorithms have been implemented in discrete-time. This was an underlying philosophy of the work done in this dissertation, i.e., plant models developed in continuous time and algorithms in discrete time, in order to have operating conditions similar to those occurring in real plants.

Tab. 5.1 - Parameters for the three-tank benchmark.

Parameter	Value
a_z	1
A	0.0154 m^2
g	9.81 m / s^2
S	3.6×10^{-5} m^2
Q_{max}	0.1×10^{-3} m^3 / s
h_{max}	0.6 m

5.4 Digital Controllers

During the research work several digital controllers were implemented and tested on different plant models and real setups. The controllers tested are the classical PID controller, the fuzzy controller and the adaptive optimal linear quadratic Gaussian controller (LQGC). A brief description of each controller is given next.

The classical linear Proportional-Integral-Derivative (PID) controller is by far the most common control algorithm used in industry (Astrom & Hagglund, 2000a; Jussila, 1992). The incremental PI version has been implemented here. For low order systems, a pole-placement approach has been used for tuning the parameters (Astrom & Hagglund, 1988). The paper by Astrom & Hagglund (2000b) describes a collection of systems suitable for testing PID controllers.

The Fuzzy controller was used in some experiments to deal with nonlinear systems. Fuzzy Control has encountered great interest in industrial applications over the past few years, and also among manufacturers of control equipment (Driankov, et al., 1996; Yager & Filev, 1994; Farinwata, et al., 2000). A discrete time fuzzy PI controller based on the Mandani type of inference has been implemented. The general architecture can be found in Palma et al. (2005d) and Ramos (1998), inspired by the PID-like fuzzy knowledge based controller (Driankov, et al., 1996). A simplified version of the fuzzy PID controller using a common rule base for the two components PI and PD was used in some experiments of this work, but

not reported in this document. Intelligent nonlinear controllers are those based on intelligent computational techniques like fuzzy logic and neural networks. Neural controllers were not used in this dissertation, but they can be a good solution especially when dealing with nonlinear systems (Narendra & Parthasarathy, 1990).

For a restricted class of faults, fault tolerance can be reached by means of control methods based on robust control and/or adaptive control (Blanke, et al., 2003). For this reason, in this work, an adaptive optimal linear quadratic Gaussian controller (LQGC), in the polynomial form, was used to control nonlinear plants. If the ARX model parameters are estimated on-line, the LQGC acts as an adaptive optimal LQG controller, and hence can be applied to nonlinear and time-varying systems. The design of the polynomial version of the LQG controller can be found in the book written by Lewis (1996). It is briefly reviewed here. Let us consider a SISO dynamic system described by the polynomial form:

$$A(z^{-1})\, y(k) = z^{-d}\, B(z^{-1})\, u(k), \qquad \text{Eq. 5.8}$$

where $y(k)$ is the output signal and $u(k)$ is the control input. The system delay is denoted d. In this discrete time formulation z^{-1} denotes the unit delay. Let us assume that the denominator polynomial is expressed by $A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_n z^{-n}$, and the numerator by $B(z^{-1}) = b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}$. The performance index $J(k)$ for this LQGC is given by:

$$J(k) = (P\, y(k+d) - Q\, w(k))^2 + (R\, u(k))^2 \qquad \text{Eq. 5.9}$$

In Eq. 5.9, the weighting polynomials $P(z^{-1})$, $Q(z^{-1})$ and $R(z^{-1})$ are design parameters selected by the engineer, and $w(k)$ is a reference or command signal. The tracking problem may be solved by selecting $P = Q = 1$, and $R = r_0$; a delayed version of the output signal $y(k)$ tries to follow a reference input signal $w(k)$. The polynomial tracker is easy to implement and it is causal, in contrast to the state-space LQR tracker where a non-causal feed-forward signal is needed. To obtain the optimal control action $u(k)$ that minimizes the performance index in minimum phase systems (e.g., all roots of $B(z^{-1})$ stable) it is necessary to solve the Diophantine equation $1 = A\, F + z^{-d}\, G$ for the intermediate polynomials $F(z^{-1})$ and $G(z^{-1})$. In terms of the Diophantine equation solution, the optimal control sequence is given by the equation:

$$(P\,B\,F + (r_0 / b_0)\,R)\,u(k) = -\,P\,G\,y(k) + Q\,w(k). \qquad \text{Eq. 5.10}$$

The optimal LQG polynomial controller is called a two-degrees-of-freedom regulator, because it has feedback and feed-forward components. Such a controller can influence the closed-loop poles as well as zeros. The LQGC requires full state feedback because the complete state is given by $y(k), y(k\text{-}1), \ldots, y(k\text{-}n), u(k\text{-}d), \ldots, u(k\text{-}(d+m))$. A polynomial self-tuning regulator (STR) can be obtained directly from the LQGC if the design parameter r_0 assumes the zero value. If the ARX parameters are estimated on-line (the typical situation), the LQGC acts as an adaptive optimal LQG controller, and hence can be applied to nonlinear and time-varying systems. The design parameter r_0 in this work has been selected in order to guarantee closed-loop stability and also good persistent excitation conditions.

Some faults cause abrupt changes on the ARX model parameters used by the adaptive optimal LQG controller. In order to minimize the possibility of loss of control due to the abrupt changes of the controller gains, it is recommended to low pass filter the LQG controller gains. In this work, in saturation situations, a small dither signal is added to the saturated signal, in order to guarantee the persistent excitation conditions.
In some cases the LQG controller can act as a fault tolerant controller (FTC) accommodating some faults effects. On systems subject to hardware reconfiguration, in some situations, the LQG controller can adapts to new system dynamics (Palma, et al., 2003b).
In the process start-up, and on some faulty situations, a non-adaptive controller must be used. The principal reason is that the estimated model is not good enough, and for these cases an adaptive controller cannot guarantee the stability of the closed-loop system, in some faulty cases.

In this work, the stability of switched systems was not investigated, but is certainly an important pointer for future research. Stability properties of a switched system in general depend on the switching signal. A switched system is stable if all individual subsystems are stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch (Liberzon, 2003). This stability subject is particularly important when dealing with adaptive systems, like the case in some experiments carried out in this work where an adaptive optimal LQG controller is used. In this work, the LQG controller gains have been low pass filtered in order to avoid abrupt transitions, and also to try to prevent the loss of the plant controllability.

5.5 Comparison of Parameter Estimation Algorithms

In this experiment three parameter estimation algorithms are compared. The algorithms are the sliding window least squares (SW-LS), the exponentially weighted recursive version of least squares (EW-RLS), and the sliding window PCR (SW-PCR). The least squares (LS) algorithm can be found in section 2.3.7, and the adaptation to a sliding window version is straightforward. The EW-RLS algorithm, with time horizon $T_0 = 1 / (1 - \lambda)$ and a forgetting factor λ, can be found in (Mosca, 1995). The SW-PCR algorithm is described in section 2.3.7. To test the algorithms a first order continuous time model with transfer function given by

$$G_0(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \qquad \text{Eq. 5.11}$$

is used, considering $K = 1$ and $\tau = 1$ s. A discretization by the zero-order-hold (ZOH) method will give an ARX(n_a=1, n_b=1, n_d=1) model of the form

$$y(k) = -a_1\, y(k\text{-}1) + b_1\, u(k\text{-}1) + e(k). \qquad \text{Eq. 5.12}$$

For a sampling period of $T_s = 0.11$ s, the model parameters are given by $[a_1\ b_1] = [-8.96\times10^{-1}\ 1.04\times10^{-1}]$. A sliding window of length 10 s was used for all the algorithms, corresponding to 91 samples. The identification is done in closed-loop, and the active controller is a PI controller (incremental version) with gains $K_p = 0.4$ and $T_i = 0.4$ s (Astrom & Hagglund (1988). The reference signal assumes the value 0.5, and a white noise dither with variance 1×10^{-3} was added. A white noise with variance 1×10^{-6} was also added to the sensor signal.

The first experiment shows the results for nominal operating conditions. An experiment with a duration of 400 s was performed and the estimates of the model parameters of ARX(n_a=1, n_b=1, n_d=1) are shown in Fig. 5.5. The sliding window algorithms SW-LS and SW-PCR show the same good performance, i.e., the estimated parameters (blue lines) are almost equal and have good convergence properties. The recursive EW-RLS algorithm presents the worst convergence (green lines) and the greatest MSE estimation errors, as

described later. The nominal model parameters $[a_1\ b_1] = [-8.96\times10^{-1}\ 1.04\times10^{-1}]$ are shown in solid black lines.

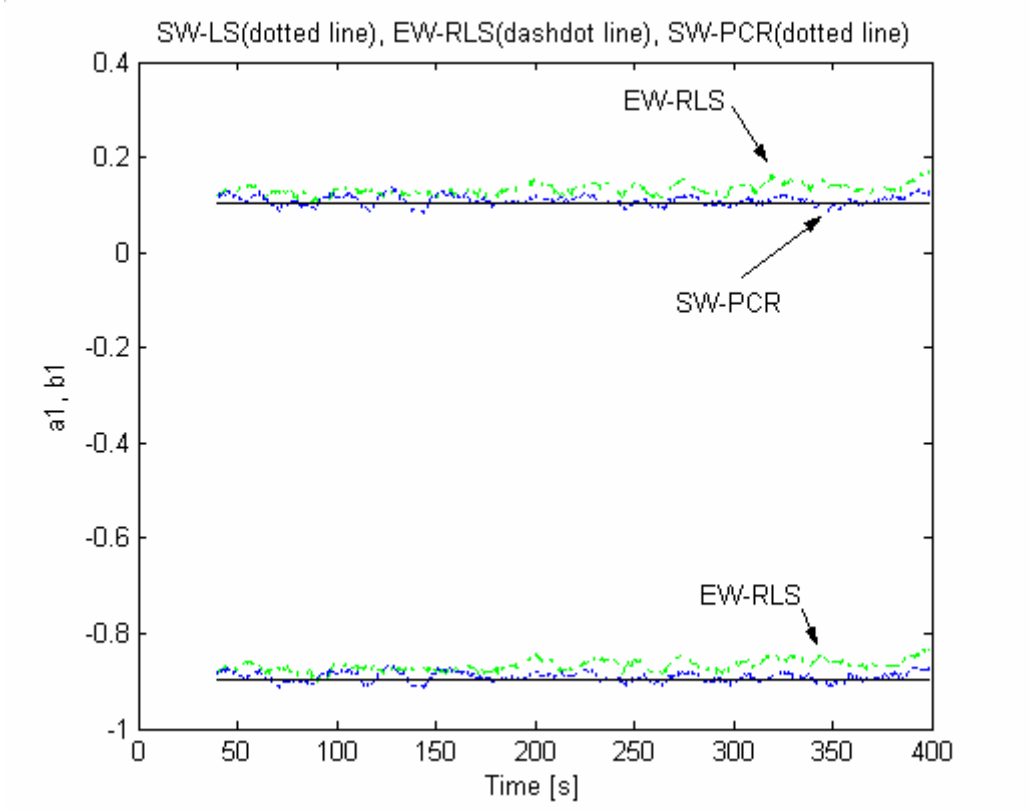


Fig. 5.5 - Estimated parameters for EW-RLS and SW-PCR algorithms.

The mean of squared errors (MSE) function is used here to compare the various parameter estimation algorithms. The MSE is a performance function given by

$$\Gamma_{\mathrm{MSE}} = \frac{1}{m}\sum_{i=1}^{m} e_i^2, \qquad \text{Eq. 5.13}$$

where e_i is the prediction error, and m is the number of samples.

For nominal operation, without faults, Tab. 5.2 shows the obtained MSE errors. The samples corresponding to the first 40 s of the experiment were not considered for the MSE calculus. It is clear that the MSE error for the EW-RLS algorithm is around eight times greater than the MSE error for the SW-PCR algorithm.

Tab. 5.2 - Comparison of MSE errors for algorithms EW-RLS and SW-PCR.

	a_1 parameter	b_1 parameter
EW-RLS algorithm, Γ_{MSE}	9.31×10^{-4}	9.31×10^{-4}
SW-PCR algorithm, Γ_{MSE}	1.19×10^{-4}	1.19×10^{-4}

When a fault occurs, a great degradation on the performance of EW-RLS algorithm happens, as shown in Fig. 5.6. The fault occurred is a change on the static gain from $K = 1$ to $K = 2$ at time instant $t_k = 240$ s. For the sliding window algorithms, SW-LS and SW-PCR (blue lines), the transient following a parameter change (jump) lasts exactly 10 s, i.e., the duration of the window length. For the EW-RLS algorithm (green lines) the fault transient is around 70 s, i.e., a value seven times greater.

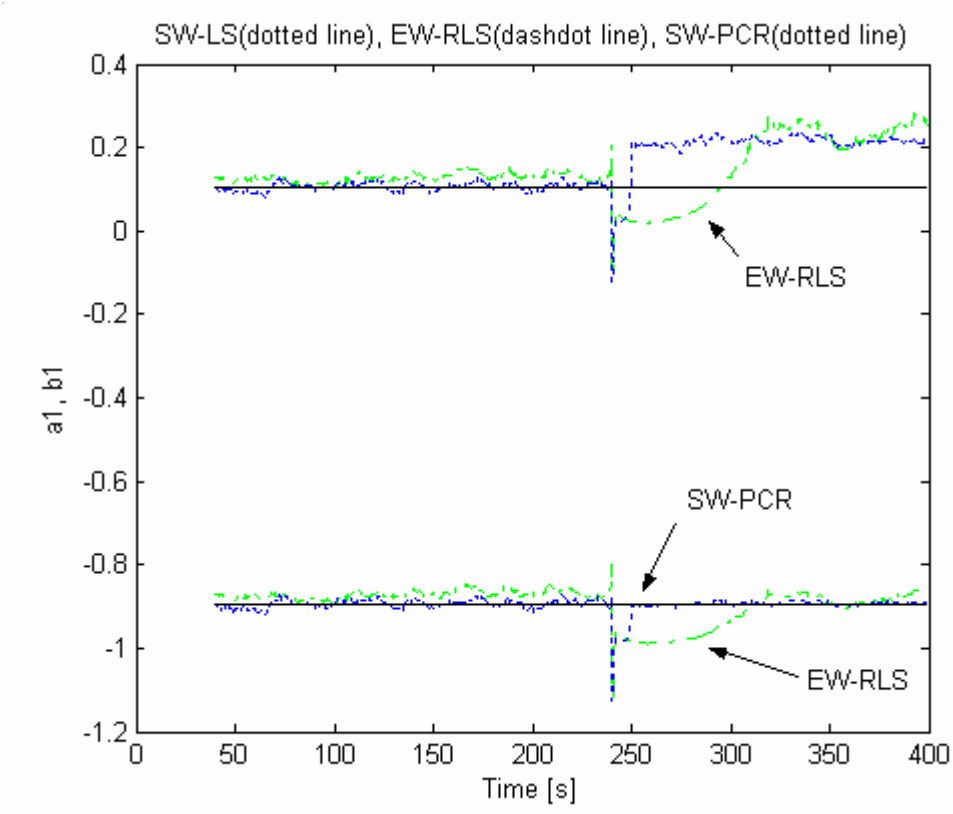


Fig. 5.6 - Performance of EW-RLS and SW-PCR algorithms for a faulty situation.

Remarks.

According to Wise & Ricker (1990) parameterized models identified by classical least-squares (LS) are generally as good as models identified by PCR. This fact has been confirmed in the experiments carried out in this work for the sliding window versions of LS and PCR,

and also in this example. The exception to this is when conditions are extremely adverse, e.g. there is poor input excitation in the data and the noise level is very high.

For fault detection and diagnosis tasks it is advantageous to use sliding window parameter estimation algorithms instead of recursive algorithms (Gertler, 1998). This fact was confirmed in the experiments done in this work.

With a sliding window of length w, the transient following a parameter change (jump) lasts exactly w-1 samples. Therefore, any isolation decision has to be delayed by w-1 samples following the detection of a change, but one may be certain that there is no spreading of the parameter jump to the estimates of other parameters.

The length w of the sliding window plays a crucial role. The variance of the parameter estimates increases with decreasing window length. A shorter window is clearly advantageous from the point of view of isolation delay, but a drawback is higher noise sensitivity as the window length decreases.

The great advantage of the sliding window (SW-PCR, SW-LS, etc) algorithms over the recursive algorithms (EWM-RLS, etc) is the fast adaptation after fault occurrence, and this is the main reason why the SW-PCR algorithm was used in this work. A drawback of the sliding window algorithms is a heavier computational load compared to the recursive algorithms.

5.6 Normality Tests

The normal distribution, also called Gaussian distribution, is a probability distribution of great importance in many fields, such as statistical data analysis, fault detection and diagnosis, medicine, psychology, biology, financial studies, etc. The normal distribution is probably the most important distribution in both the theory and application of statistics. A definition of its probability distribution function was given in section 2.3.3.

As a consequence of the central limit theorem we know that regardless of the distribution of the data (population), the sampling distribution of the sample mean is approximately normal. So it is expected that, for a stationary dynamic process under nominal operation, some stochastic signals (features and residuals) used for fault detection and diagnosis purposes follow approximately a normal distribution.

In this experiment, some tests are applied to evaluate if the ARX model parameters estimated on-line approximately obey a normal distribution. Considering a plant model in continuous time with a transfer function expressed by

$$G_0(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \qquad \text{Eq. 5.14}$$

with a static gain $K = 1$ and a time constant $\tau = 1$ s. A discretization by the zero-order-hold (ZOH) method gives an ARX(n_a=1, n_b=1, n_d=1) model of the form

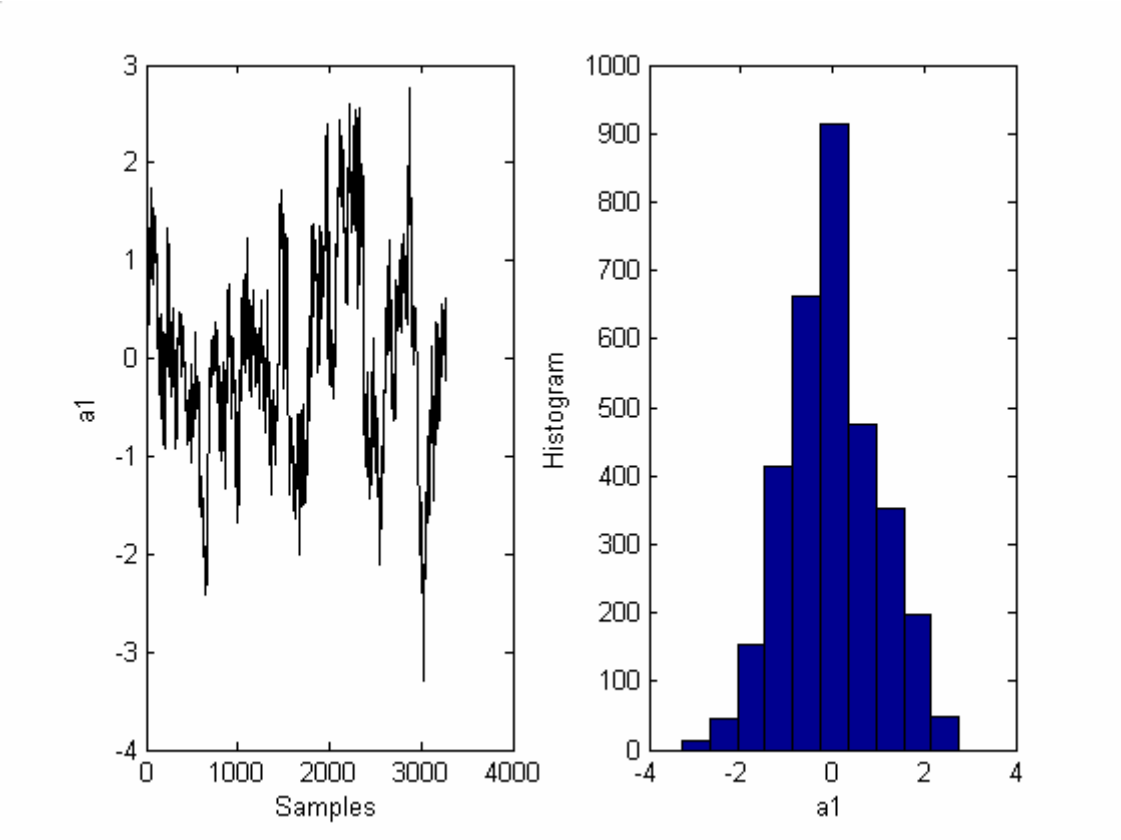
$$y(k) = -a_1\, y(k\text{-}1) + b_1\, u(k\text{-}1) + e(k). \qquad \text{Eq. 5.15}$$

For a sampling period of $T_s = 0.11$ s, the model parameters are given by $[a_1\ b_1] = [-8.96\times10^{-1}\ 1.04\times10^{-1}]$. The identification is carried out in closed-loop, and the active controller is a PI controller (incremental version) with gains $K_p = 0.4$ and $T_i = 0.4$ s (Astrom & Hagglund (1988). The reference signal assumes the value 0.5, and a white noise dither with variance 1×10^{-3} was added. A white noise with variance 1×10^{-6} was added to the sensor signal. An experiment with duration of 400 s was performed and the model parameters of an ARX(n_a=1, n_b=1, n_d=1) were computed. A sliding window of length 10 s, corresponding to 91 samples, was used by the SW-PCR parameter estimation algorithm (described in section 2.3.7).

To test the normality of the ARX model parameters three tests have been performed using the Matlab available functions on toolboxes: a) the normal probability (rankit) plot; b) the Lilliefors test; c) the D'Agostino-Pearson omnibus test.

Results shown here are for the estimated model parameter $a_1(k)$. The data has been auto-scaled in order to guarantee a zero mean and unitary variance. Similar results are obtained for the parameter $b_1(k)$. Fig. 5.7.a) shows the parameter $a_1(k)$, i.e., the data. The first 40 samples of the experiment are not shown, and were not considered since they contain transient start-up data. The data histogram appears in Fig. 5.7.b), where it can be observed that the distribution does not deviate significantly from a symmetric bell-shaped curve typical of a normal distribution.

Fig. 5.8 depicts the normal probability (rankit) plot. The rankit plot is a graphical method for determining whether sample data conform to a normal distribution, based on a subjective visual examination of the data. If the data comes from a normal distribution, the plot will appear linear. Other probability density functions will introduce curvature in the plot. Since the plotted data points (marked with "+") do not deviate significantly from the straight line, the data is approximately normally distributed.



a) Data, parameter a1. b) Histogram

Fig. 5.7 - Data and histogram.

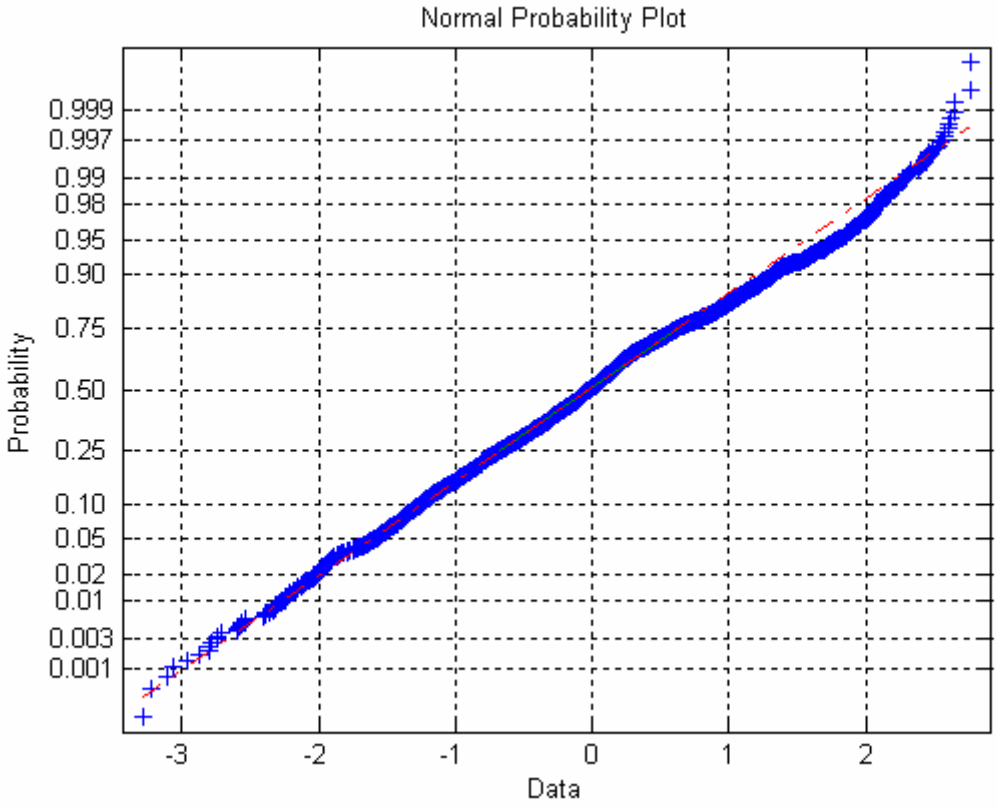


Fig. 5.8 - Rankit plot for parameter a1.

The Lilliefors test of normality is a modification of the Kolmogorov-Smirnov test, and can be found in (Conover, 1999). The null hypothesis, $H = 0$, of this test is that the data is normally distributed, considering the result given by H. If $H = 1$ we can reject the null hypothesis, i.e., the data is not normally distributed. This normality criterion was also applied to the data, considering an α-level (significance level, probability associated with the criterion) of 5 %, and the result was that the data is not normally distributed.

Some authors argue that the D'Agostino-Pearson omnibus test presents, in most cases, a better performance than the methods based on the Kolmogorov-Smirnov test. The D'Agostino-Pearson omnibus test for assessing normality of data using skewness and kurtosis can be found in (Zar, 1999). In this experiment an algorithm written by Trujillo-Ortiz and Hernandez-Walls was used; the algorithm can be found on the site http://www.mathworks.com. The D'Agostino-Pearson omnibus test, for a significance level of 5 %, indicates that the data is normally distributed. The skewness is a measure of the asymmetry of the data around the sample mean. The skewness of the normal distribution (or any perfectly symmetric distribution) is zero. To the estimated parameter a_1 the skewness is 7.4×10^{-2}. The kurtosis is a measure of the shape of the distribution, indicating how outlier-prone a distribution is. The kurtosis of the normal distribution is 3. To the estimated parameter a_1 the kurtosis is 2.9.

Different normality tests can give different results, as shown in the experiment. As a conclusion, different analytical tests must be performed in order to draw conclusions about the normality of data, and to take decisions about the type of statistics used for process monitoring. It is certain that the statistical properties of the ARX model parameters depend on the input persistent excitation conditions, and also on the performance of the parameter estimation algorithm.

PCA models have the advantage that the scores variables produced which are linear combinations of the original variables, are more normally distributed than the original variables themselves. This is a consequence of the central limit theorem. Thus, we would expect the scores, which are a weighted sum like a mean, to be approximately normally distributed (Gnanadesikan, 1997; Wise and Gallagher, 1996). If there is enough data in the training set to capture the normal process variations, the *T2* statistics can be an effective tool for process monitoring, even if there are smooth deviations from the normality or statistical independence assumptions. In section 5.8, a two dimensional scores space in Fig. 5.21 is shown for the case of data variables given by ARX model parameters, for a dynamic process

under nominal operation. The scores are approximately normally distributed. Since the data does not obey a perfectly normal distribution and the relations between data (model parameters) are not perfectly linear, the value T_α^2 must be adjusted experimentally, in order to obtain a desired rate of false alarms.

Remarks.
The assumption of normality for the data is particular important to compute thresholds used for fault detection and fault diagnosis. In this work, the assumption of normality for the FDD features signals, under nominal operating conditions, is quite reasonable. To reduce the false alarm rates, in this work, the fault detection and isolation signals are low pass filtered. If necessary the thresholds obtained assuming the data is normally distributed can be experimentally adjusted to guarantee a reasonable FDD performance in terms of ratio of false alarms.

5.7 FDD in LTI Systems based on Dynamic Features of ARX Models

For single-input single-output (SISO) linear and time-invariant (LTI) systems, in section 3.3.2 a new fault detection and diagnosis (FDD) method based on dynamic features (static gain and bandwidth) of black-box ARX models has been proposed.
To compute these dynamic features, it is necessary to estimate the ARX model parameters on-line. The static gain is simple to compute. The bandwidth is computed based on the rise-time, and the rise-time is obtained from the step response of the ARX model. The sliding window SW-PCR estimation algorithm, proposed in this work, is used for on-line estimation of the ARX model parameters. The sliding window length here is 10 s, corresponding to 91 samples, for a sampling period of T_s = 0.11 s.
The on-line FDD methodology applied here has been proposed in section 3.3.2. The FDD architecture depicted in Fig. 5.9 is briefly reviewed next. Based on input-output data, $\mathbf{u}(k)$ and $\mathbf{y}(k)$, the sliding window SW-PCR algorithm estimates the ARX model parameters, $\boldsymbol{\theta}(k)$, on-line. The dynamic features of the ARX model, static gain $s_g(k)$ and bandwidth $b_w(k)$, are computed on-line based on $\boldsymbol{\theta}(k)$. These features are used for fault detection and diagnosis. The neural nonlinear discriminant analysis classifies the on-line data pattern, $[s_g(k)\ b_w(k)]$,

into a certain fault class $\mathbf{p}(k)$. In the two-dimensional space defined by the features $s_g(k)$ and $b_w(k)$, the nominal operation is characterized by a pattern (cluster), and the faults by other clusters. A fault alarm signal, $a_m(k) = 1$, is generated if the on-line feature data computed at time k, $(s_g(k), b_w(k))$, falls outside the nominal cluster, i.e., if the corresponding class is different from the nominal class associated with fault F_0. A fault detection signal, $f_d(k)$, is obtained by low pass filtering the fault alarm signal and by thresholding. The fault isolation signal, $f_i(k)$ is obtained by fault classification using a knowledge-based system, and by low pass filtering.

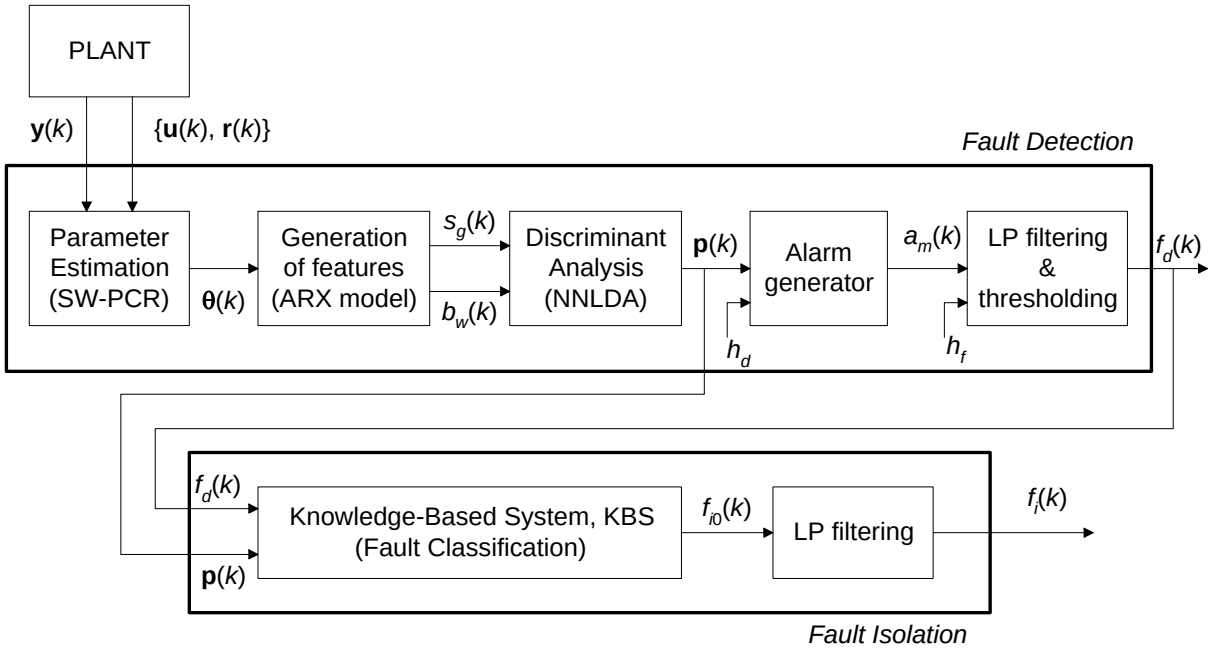


Fig. 5.9 - Architecture of the FDD approach based on dynamic features of ARX model.

Results for an experiment applying this FDD methodology to a LTI SISO system, a continuous time DC motor linear model, are presented here. The complete description of the DC motor model is given in section 5.3.2. The simplified continuous-time transfer function of the DC motor is given by a 2^{nd} order system (Kuo, 1995):

$$G_m(s) = \frac{\omega_r(s)}{u_a(s)} = \frac{K_m}{(L\,J)\,s^2 + (L\,K_f + R\,J)\,s + (R\,K_f + K_m\,K_b)}. \qquad \text{Eq. 5.16}$$

The transfer function relates the angular velocity $\omega_r(t)$ of the shaft and the applied voltage $u_a(t)$.

The description and values of each model parameter (resistance R, etc) are given in section 5.3.2. The experiments were performed using the following architecture:

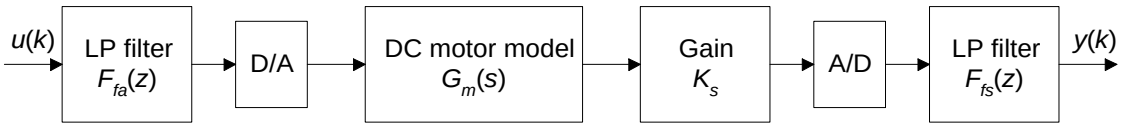


Fig. 5.10 - DC motor model architecture with filters.

The overall system shown in Fig. 5.10 has a nominal unitary static gain, since $G_m(s=0) = 1/4.1$, $K_S = 4.1$, and the low pass filters have unitary static gain. A white (Gaussian) noise with variance 1×10^{-8} has been added at the sensor output. In order to guarantee the persistent excitation conditions (PEC), a white noise dither signal with variance 1×10^{-4} was added to the reference signal $r(k)$.

In closed-loop operation, the general architecture used in this work depicted in Fig. 5.11 has been adopted here, as also in the other experiments shown in this dissertation. Here, a discrete time PI controller was used for speed control, with gains $K_p = 2.55$ and $T_i = 2.02$ s obtained using a pole placement approach (Astrom & Hagglund, 1988). All the supervision and fault detection and diagnosis algorithms have been implemented in discrete time.

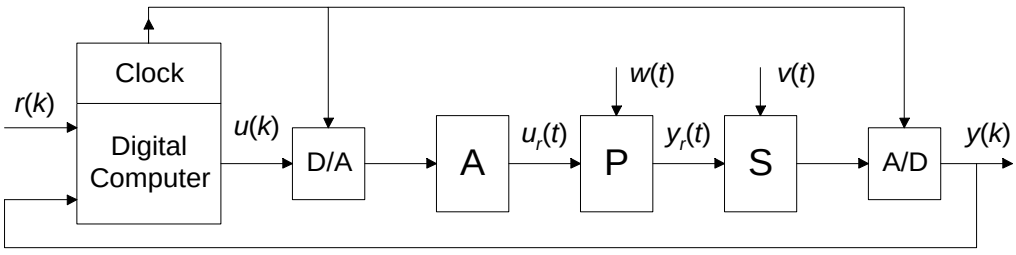


Fig. 5.11 - Closed-loop control architecture.

All the values have been scaled to the range [0; 1]. The nominal operation, corresponding to fault F_0, is assumed for a reference signal around 0.5. Parametric faults are considered here. Two faults on the motor resistance are considered in this experiment, and one fault on the output filter. Fault F_1 is an increase of 50 % in the motor resistance from $R = 2\ \Omega$ to $R = 3\ \Omega$. Fault F_2 is a decrease of 70 % in the motor resistance from $R = 2\ \Omega$ to $R = 0.6\ \Omega$. Fault F_3 is an increase of the design parameter (λ, pole location) of the output low-pass filter

$F_{fs}(z) = (1-\lambda) / (1-\lambda z^{-1})$ from 0.7 to 0.95, and the effect is a decrease of the system bandwidth. The results obtained for the experiments are shown later.

For the fault F_1, an increase of 50 % in the motor resistance from $R = 2\ \Omega$ to $R = 3\ \Omega$, Fig. 5.12 shows the input-output and FDD signals.

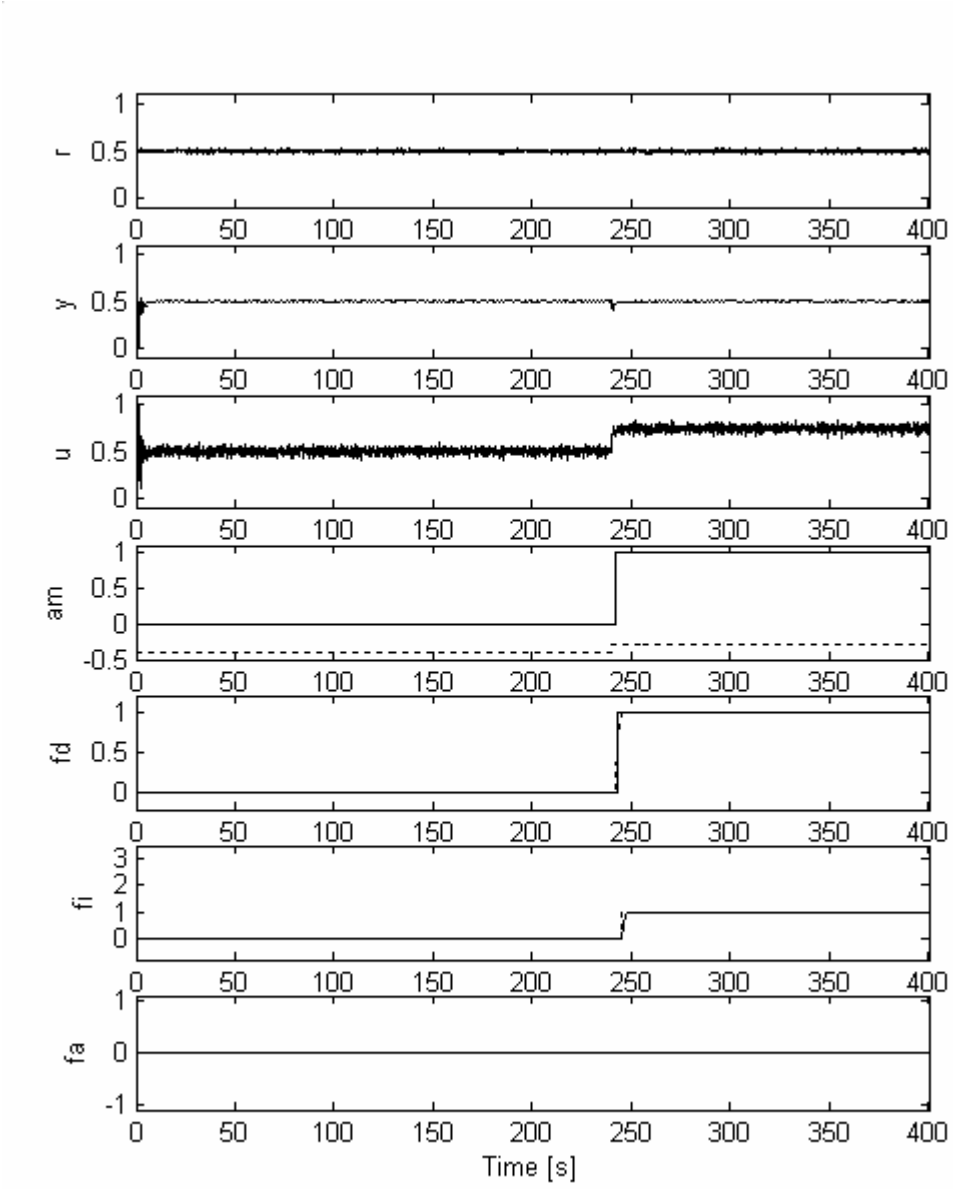


Fig. 5.12 - Input-output and FDD signals for fault F1.

The reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$ can be observed in Fig. 5.12. The fault alarm signal $a_m(k)$ and the fault detection signal $f_d(k)$ appear next. The fault isolation signal $f_i(k)$ indicates a correct isolation, assuming the value one due to fault F_1. The magnitude is not considered here, and that is the reason the fault analysis signal $f_a(k)$ is

zero. The fault occurs at time instant t_k = 240 s. The fault detection delay is 2.8 s, and the fault isolation delay is 8 s. For this fault, the PI controller is able to compensate the fault effect, since the control error is maintained around zero after the fault occurrence.

Fig. 5.13 shows the parameters (label "th-yu") of the input-output ARX(2, 1, 2) model M_{yu}, $\theta_{yu}(k)$, for a model relating the output signal $y(k)$ with the input signal $u(k)$; the nominal values are given by $[a_1\ a_2\ b_2] = [-1.76\ 7.84\times10^{-1}\ 2.48\times10^{-2}]$. The static gain s_g and the bandwidth b_w, used here as features for FDD, are computed using the parameters $\theta_{yu}(k) = [a_1(k)\ a_2(k)\ b_2(k)]$. The parameters (label "th-yr") of the reference-output ARX(2, 1, 2) model M_{yr}, $\theta_{yr}(k)$, are also shown; the nominal values are given by $[a_1\ a_2\ b_2] = [-1.79\ 8.56\times10^{-1}\ 6.18\times10^{-2}]$. The model M_{yr} relates the output signal $y(k)$ and the reference signal $r(k)$. This fault causes changes on the parameters of both models, so both models can be used for FDD purposes. Finally, the fault activation (trigger) signal $f_{as}(k)$ is depicted.

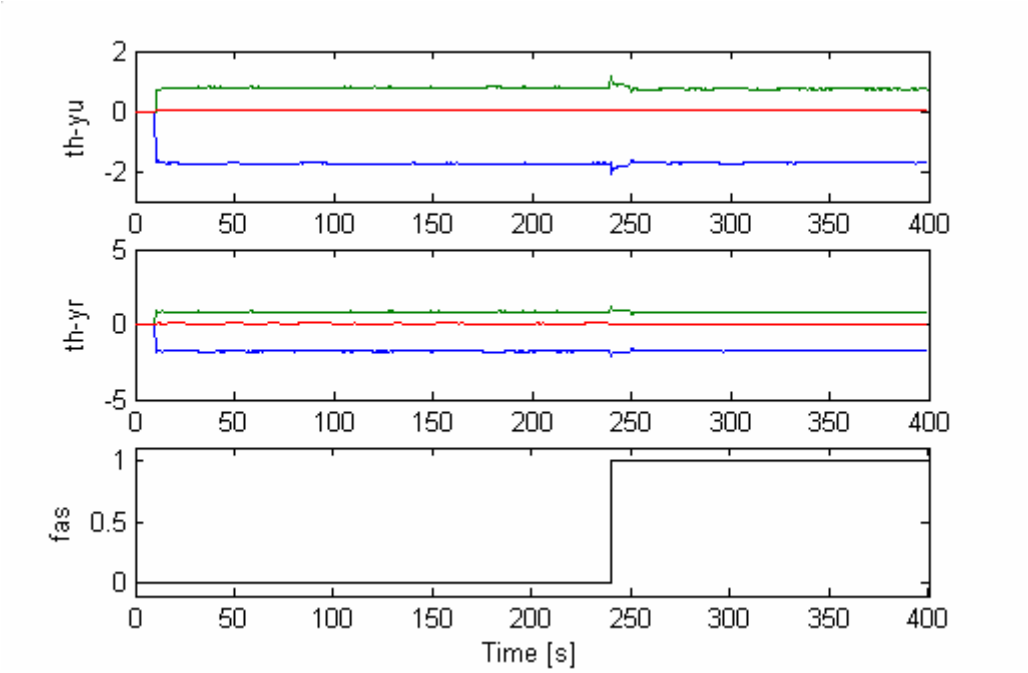


Fig. 5.13 - Evolution of ARX model parameters.

In Fig. 5.14 the parameters of the input-output ARX model, M_{yu}, are depicted. Next, the static gain and the bandwidth appear. Finally, the two dimensional features space appears where the fault trajectory from cluster F0 to cluster F1 can be observed. The clusters centres are shown.

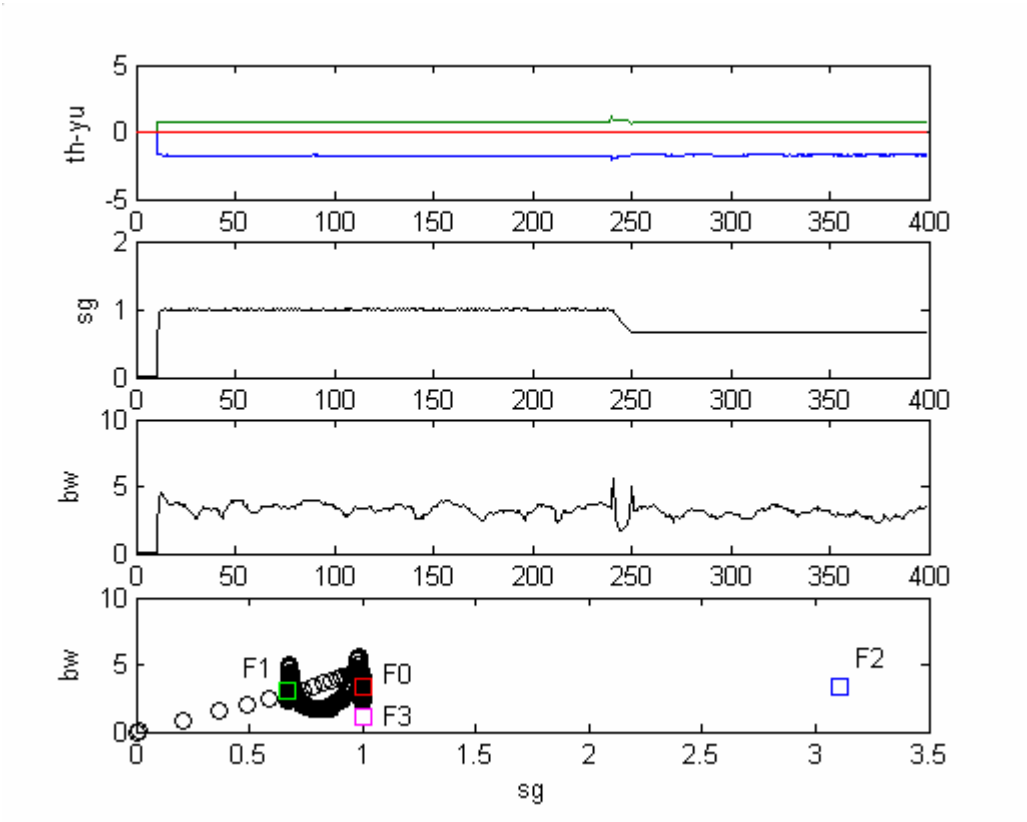


Fig. 5.14 - Features (static gain and bandwidth) for FDD.

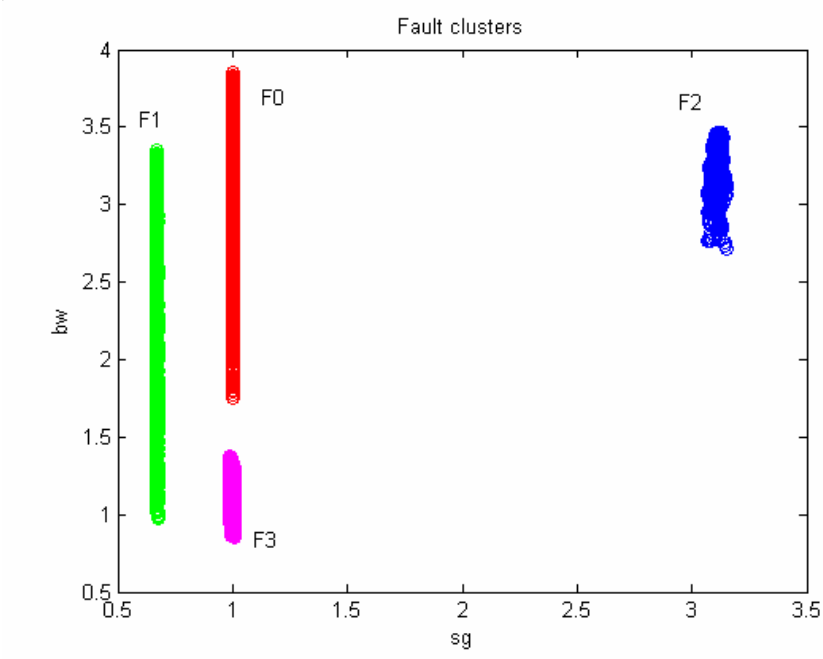


Fig. 5.15 - Fault clusters used for NN training of neural NNLDA.

In Fig. 5.15, the patterns (clusters), for each fault, used for training the neural network that implements the neural nonlinear discriminant analysis (neural NNLDA) are shown.

A feed-forward neural network is used for implementing the neural NNLDA for fault classification. The training is done off-line for 500 epochs, using the Levenberg-Marquardt optimization algorithm. The sum squared error (*SSE*) performance function was selected, and the value obtained at the end of the 500 epochs was around $SSE = 1.1\times10^{-2}$.

In this figure, it is clear that the faults F_1 and F_2 are isolated mainly due to the effect on the static gain. For the case of fault F_3, the expected decrease of the bandwidth is confirmed. For the faults F_2 and F_3, the variance of the parameters is lower than for the faults F_0 and F_1.

Looking again at Fig. 5.15, some regards can be drawn about fault detectability and fault isolability. If some fault cluster, F_i, is overlapped or adjacent with the nominal fault cluster, F0, then the fault F_i cannot be detected very well. This situation also causes problems in the neural network training associated with the neural NNLDA, since the training algorithm cannot guarantee a reasonable performance, i.e., the sum squared error (*SSE*) stays high. Consequently, the neural NNLDA cannot separate the fault patterns very well. A similar situation occurs, for the isolation task, when clusters associated with faults F_i and F_j are overlapped or near. Both fault detection and fault isolation tasks are influenced by the noise variance and by the persistent excitation conditions, since both tasks depends on the on-line estimated ARX parameters, $\theta(k)$.

Now, the results obtained with fault F_2, a decrease of 70 % in the motor resistance from $R = 2\ \Omega$ to $R = 0.6\ \Omega$ are presented.

Fig. 5.16 shows the reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. The other graphs show the fault alarm signal $a_m(k)$ and the fault detection signal $f_d(k)$.

The fault isolation signal $f_i(k)$ indicates a correct isolation, assuming the value two due to fault F_2. The magnitude is not considered here, and that is the reason the fault analysis signal $f_a(k)$ is zero.

The fault occurs at time instant $t_k = 240$ s. The fault detection delay is 0.8 s, and the fault isolation delay is 12 s. The fault isolation task, using the sliding window SW-PCR estimation algorithm, can only guarantee a correct value after the window length of the SW-PCR algorithm, i.e., for this case after at least 10 s.

For this fault, the output shows some oscillations, since the PI controller with fixed gains is not very well tuned for this faulty situation. Also, a situation of actuator saturation occurs in certain time instants.

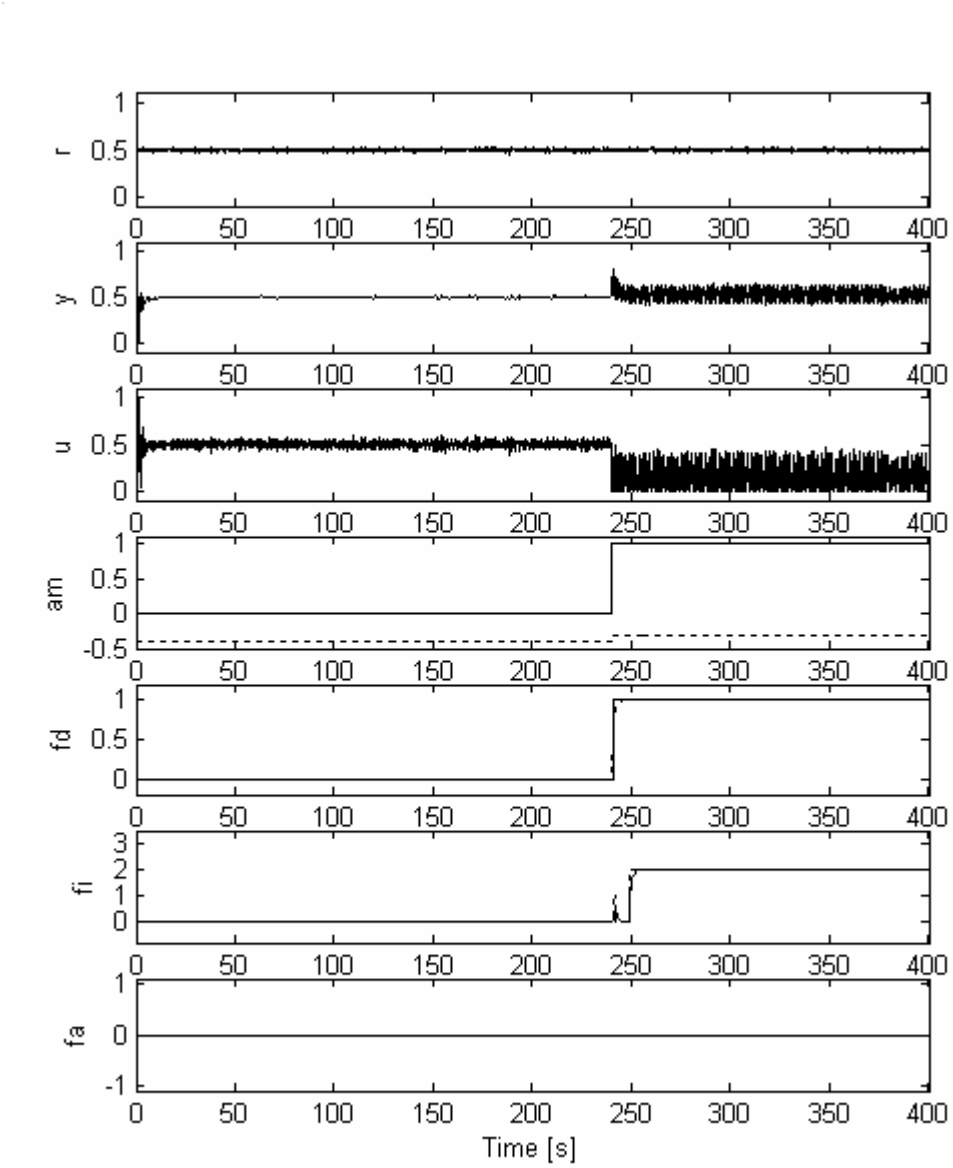


Fig. 5.16 - Input-output and FDD signals for fault F2.

Fig. 5.17 shows, from top to bottom, the ARX model parameters (label "th-yu"), the static gain as a function of time, and the bandwidth as a function of time. Finally, the two dimensional features space formed by static gain $s_g(k)$ and bandwidth $b_w(k)$ appears, where the fault trajectory from cluster F0 to cluster F2 can be observed.

As seen, when the fault occurs, the estimated static gain increases from 1 to approximately 3.2. The variation on the mean value of bandwidth is insignificant, but the variance value decreases significantly.

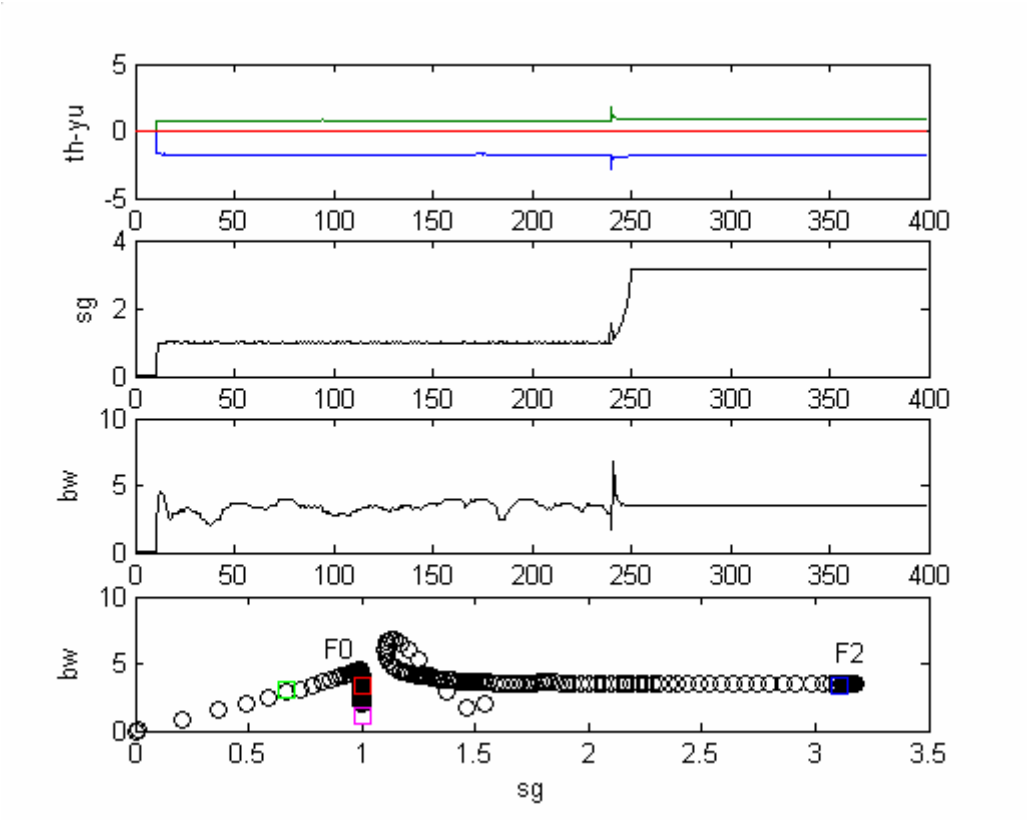


Fig. 5.17 - Features (static gain and bandwidth) for FDD.

Finally, the results for the fault F_3 are presented next.

The fault F_3 corresponds to an increase of the design parameter (λ, pole location) of the output low-pass filter $F_{fs}(z) = (1-\lambda) / (1-\lambda z^{-1})$ from 0.7 to 0.95. This fault is very difficult to detect looking only at the input-output signals. A decrease of the system bandwidth is expected, since the pole location moves from 0.7 to a higher value near 1.

Fig. 5.18 shows the reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. The other graphs show the fault alarm signal $a_m(k)$ and the fault detection signal $f_d(k)$. The fault isolation signal $f_i(k)$ indicates a correct isolation, assuming the value three due to fault F_3. The magnitude is not considered here, and that is the reason the fault analysis signal $f_a(k)$ is zero in this experiment.

The fault occurs at time instant $t_k = 240$ s. The fault detection delay is 8.2 s, and the fault isolation delay is 13 s.

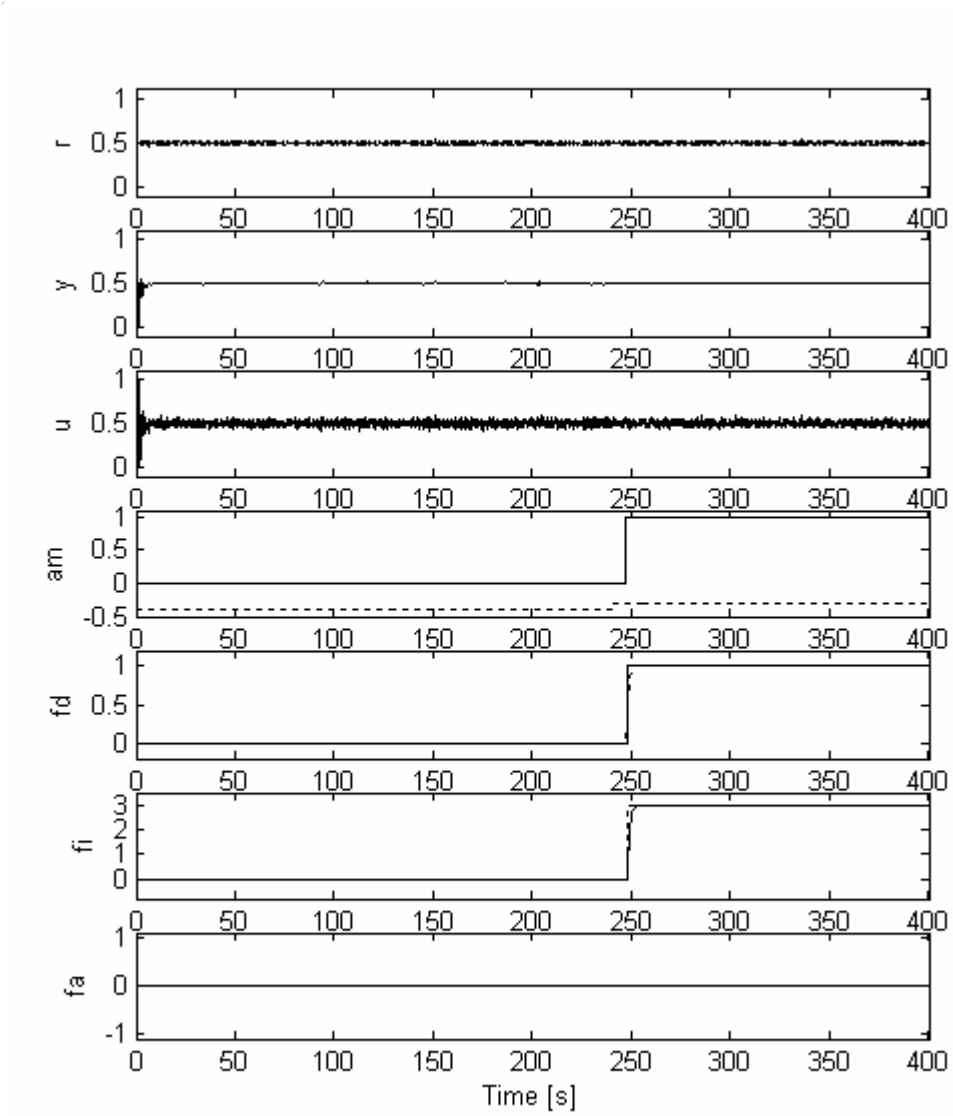


Fig. 5.18 - Input-output and FDD signals for fault F3.

As observed in Fig. 5.18, this fault is particularly difficult to detect and diagnose, since the symptoms are not visible looking only at the input-output signals.

This FDD approach is able to detect and diagnose this type of parametric fault, since the symptoms are changes on the parameters of the ARX model, as depicted in Fig. 5.19 (label "th-yu"). Since the filter $F_{fs}(z)$ has a unitary static gain, the fault can only be isolated via the effect on the bandwidth (feature), as observed in the graphs of Fig. 5.19.

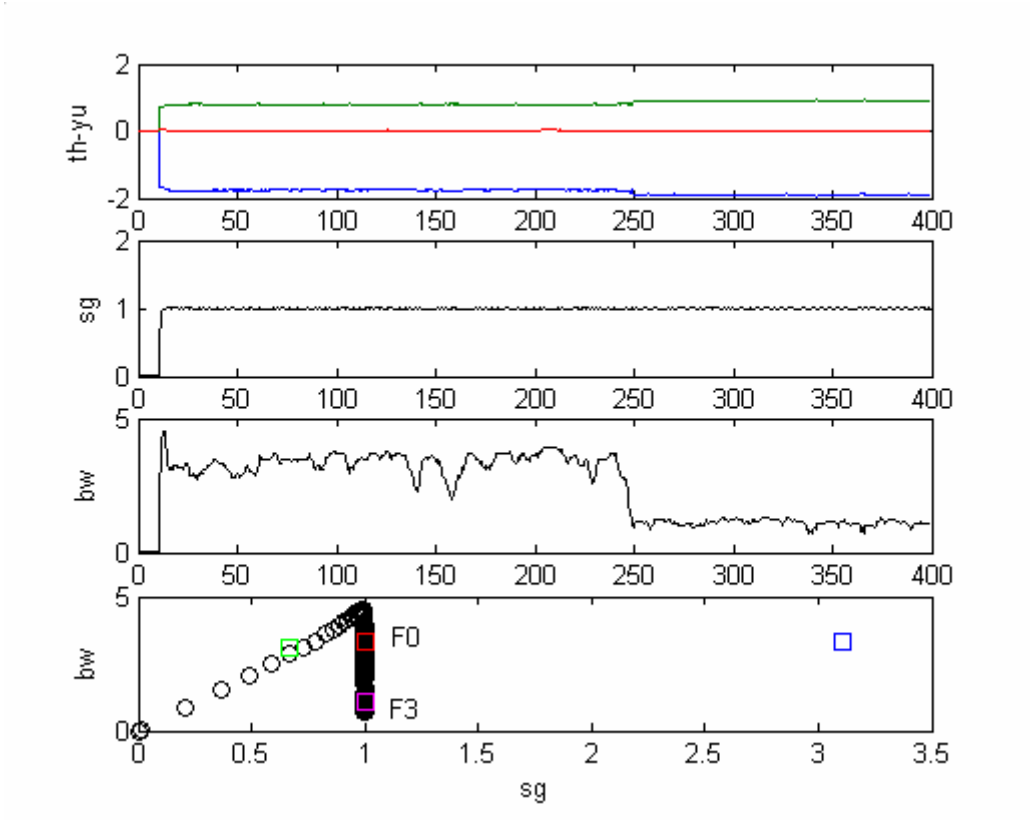


Fig. 5.19 - Features (static gain and bandwidth) for FDD.

Remarks.

The experimental results obtained with this proposed on-line fault detection and diagnosis (FDD) approach, based on dynamic features of ARX SISO models, show a good performance for the class of faults tested. In fact, the static gain and the bandwidth computed on-line based on ARX models are two features that can be used for fault detection and diagnosis tasks.

The parameter estimation of ARX models plays a crucial role here, since both fault detection and diagnosis tasks depend on it. The sliding window PCR algorithm presents a good performance, since the estimates are consistent.

The FDD approach requires good persistent excitation conditions, in order to obtain a good accuracy for the parameter estimates and small parameters variances.

Here, the FDD approach was applied using an input-output ARX model M_{yu} relating the output signal $y(k)$ and the input signal $u(k)$. Nevertheless, this FDD method can also be based on a reference-output ARX model M_{yr}, relating the output signal $y(k)$ and the reference signal $r(k)$.

This FDD approach was proposed in Chapter 3 (section 3.3) to deal with linear systems. In fact, it can be applied also to nonlinear systems, since the on-line parameter identification allows an on-line adaptation of the ARX model to the plant changes.

5.8 Combined FDD Approach for Linear Systems based on PCA & IMX

A new on-line combined fault detection and diagnosis (FDD) approach for linear systems was proposed in section 3.6. The control architecture is here again depicted, in Fig. 5.20, and used in this experiment.

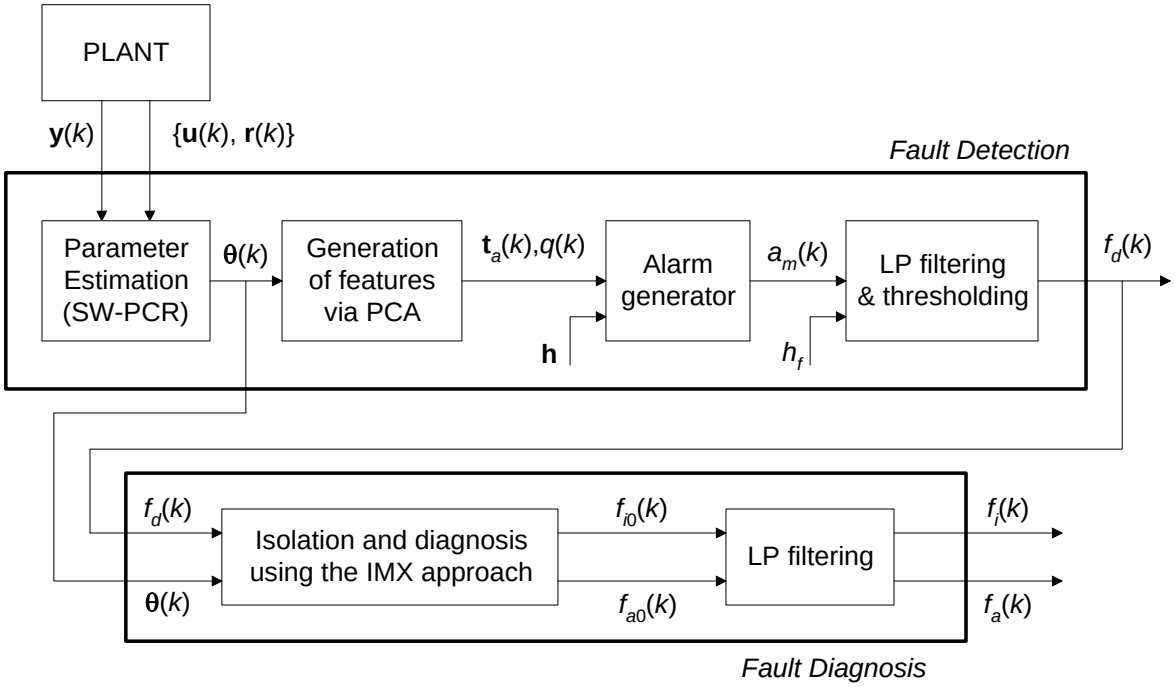


Fig. 5.20 - Architecture of the combined FDD approach (FDD-PCA-IMX).

The fault detection is achieved via the application of principal component analysis (PCA) to the parameters $\boldsymbol{\theta}(k)$ of an ARX model. The sliding window PCR algorithm is used to estimate $\boldsymbol{\theta}(k)$ on-line. PCA generates the features, two dimensional scores $\mathbf{t}_a(k)$ and square of prediction error $q(k)$, for fault detection. The fault diagnosis is based on the influence matrix (IMX) method using the Jacobian of the model parameter vector, $\boldsymbol{\theta}$, with respect to the physical parameters, $\boldsymbol{\gamma}$.

Some experimental results obtained for an application of the FDD methodology to a continuous time DC motor model used in the section 5.7, and described in more detail in section 5.3.2, are presented here.

This combined fault detection and diagnosis methodology use the ARX model parameters as features for FDD, and can be applied using input-output ARX models M_{yu} as described in Chapter 3 (Example 5), or based on reference-output ARX models M_{yr}. In the experiments shown here, a reference-output ARX(2, 1, 2) model, M_{yr}, relating the output signal $y(k)$ and the reference signal $r(k)$, is used. Using reference-output ARX models, both faults on the plant and on the controller can be detected and diagnosed. The parametric faults under consideration in this experiment are a fault on the motor resistance R and a fault on the proportional gain K_p of the PI controller.

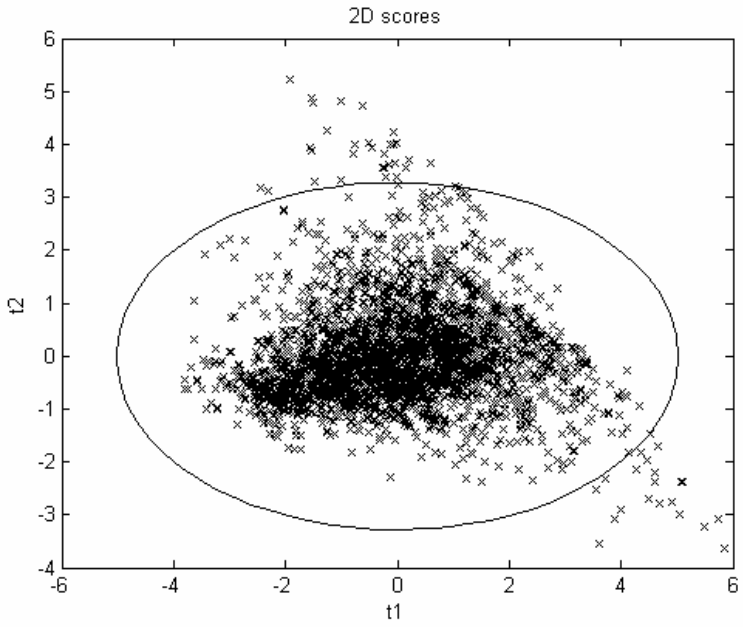


Fig. 5.21 - Two-dimensional scores space for nominal operation.

In nominal operation, an experiment is performed in order to obtain nominal data, i.e., nominal ARX model parameters, $\boldsymbol{\theta}_n$. The mean value and the standard deviation of each parameter are computed, in order to standardized the parameter values. The mean values for the ARX nominal model parameters $[a_1\ a_2\ b_2]$ are $\boldsymbol{\mu}_n = [-1.79\ \ 8.56\times10^{-1}\ \ 6.18\times10^{-2}]$. The standard deviations for each parameter are given by the vector $\boldsymbol{\sigma}_n = [1.24\times10^{-2}\ \ 6.95\times10^{-3}\ \ 9.60\times10^{-3}]$.

The PCA analysis is applied to the ARX model parameters in order to perform the fault detection task. The *T2* statistics and the *Q* statistics are used for fault detection. Fig. 5.21

shows the two-dimensional scores space obtained from data (ARX model parameters) captured during an experiment in nominal operation. For this nominal PCA model, the amount of variance explained by $a = 2$ principal components is more than 99 %. The ARX model parameters (data) have been standardized, in order to obtain a zero mean and unitary variance. Theoretically, for a multivariate normal distribution according to the *T2* statistics presented in Chapter 2, the threshold for the elliptical region in the scores space is given by

$$T_{\alpha}^{2} = \frac{a\,(n\text{-}1)\,(n\text{+}1)}{n\,(n\text{-}a)}\,F_{\alpha}(a,\, n\text{-}a). \qquad \text{Eq. 5.17}$$

Here, this theoretical threshold value for the *T2* statistics is an approximated value. In this experiment, since the relations between data (model parameters) are not perfectly linear and the data does not obey a perfectly normal distribution, the value T_{α}^{2} must be adjusted experimentally, in order to obtain a reasonable rate of false alarms. As observed in Fig. 5.21 the original data, $\theta(k)$, does not obey a perfectly normal distribution, but the deviation from normality is not significant. The value used in the experiment is $T_{\alpha}^{2} = 12.2$; this value was obtained assuming $a = 2$, $n > 120$, $\alpha = 0.05$, $F_{\alpha}(a, n\text{-}a) = 3$, and was multiplied by an empirical value ($K_f = 2$) to compensate the deviation from normality. The threshold for the *Q* statistics (square of prediction error, SPE) has been computed using a three sigma limit approach, and the values used for the mean and the standard deviation are $[\mu\ \sigma] = [6.13\times10^{-9}\ \ 8.36\times10^{-9}]$.

For fault diagnosis (isolation and analysis) the influence matrix (IMX) method is used.
The influence matrix is computed off-line based on data obtained from faulty situations. The influence matrix obtained in this experiment is given by Eq. 5.18.
The elements of the influence matrix are the slopes of the Jacobian. These slopes are computed based on data depicted in Fig. 5.22, where the nonlinear relations between model parameters and physical parameters can be observed. The values for the nominal operation are $R = 2.0\ \Omega$ and $K_p = 2.55$, and the others are values for faulty situations. The influence matrix method assumes that the model parameters are multi-linear in the physical parameters.
The theoretical multi-linearity assumption is not verified in most practical systems as observed in Fig. 5.22. The consequence is the increased difficulty of detection of small faults that have influence vectors near each other, and the degradation of the overall FDD performance.

$$\boldsymbol{\Omega} = [\frac{\partial \boldsymbol{\theta}}{\partial \gamma_1} \cdots \frac{\partial \boldsymbol{\theta}}{\partial \gamma_p}]_{\gamma = \gamma^{\text{nom}}} = \begin{bmatrix} \frac{\partial a_1}{\partial R} & \frac{\partial a_1}{\partial K_p} \\ \frac{\partial a_2}{\partial R} & \frac{\partial a_2}{\partial K_p} \\ \frac{\partial b_2}{\partial R} & \frac{\partial b_2}{\partial K_p} \end{bmatrix} = \begin{bmatrix} 4.52\times10^{-2} & 3.42\times10^{-3} \\ -7.98\times10^{-2} & 2.84\times10^{-3} \\ -3.74\times10^{-2} & 6.29\times10^{-3} \end{bmatrix} \qquad \text{Eq. 5.18}$$

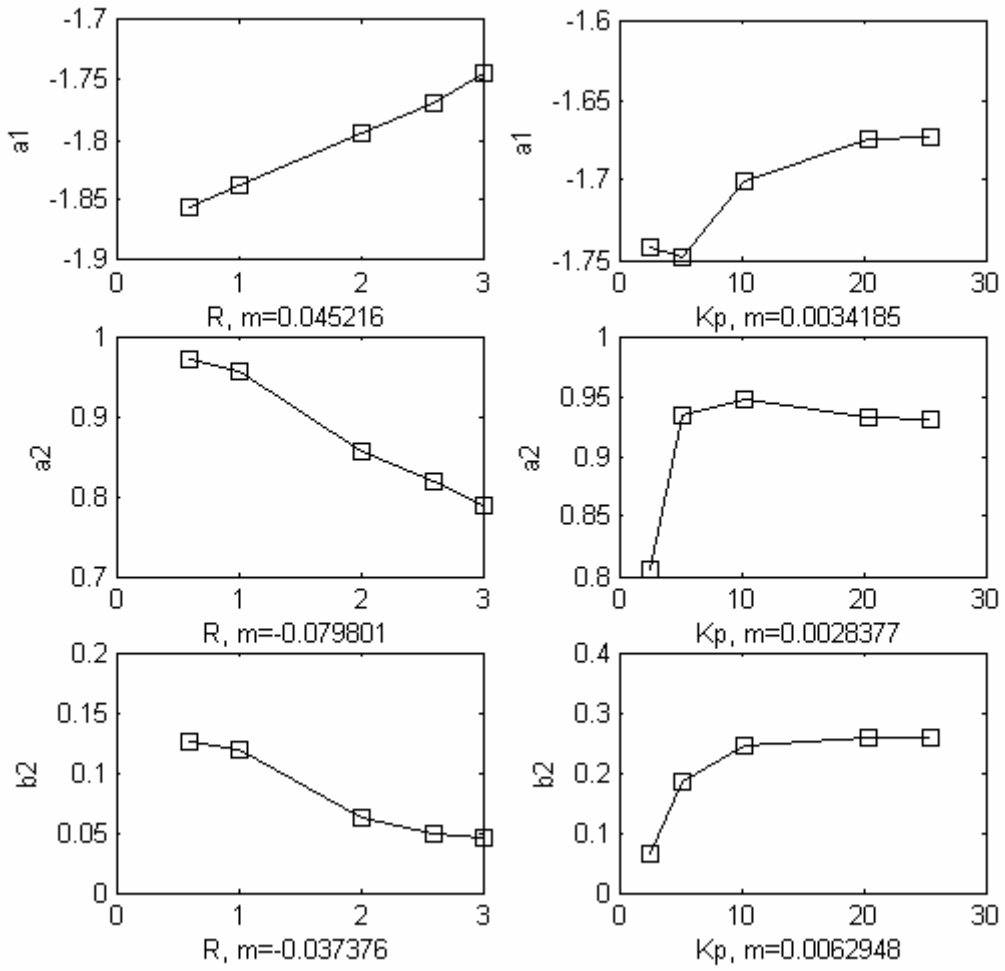


Fig. 5.22 - Influence matrix slopes, and nonlinear relations.

For the fault F_1 considered here, an increase of the motor resistance from $R = 2\ \Omega$ to $R = 3\ \Omega$, experimental results are shown next.

The fault occurs at time instant $t_k = 240$ s. In Fig. 5.23, the first graphs show the reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. Next, from top to bottom, the fault alarm signal $a_m(k)$, the trigger fault signal, the alarm associated with the *T2* statistics, and finally the alarm associated with the *Q* statistics (SPE) appear. Next, the fault detection signal

$f_d(k)$ in solid line and the respective signal without low-pass filtering and thresholding in dotted line appear. The isolation signal $f_i(k)$ appears next, with a signal that indicates an increase or a decrease of the physical faulty parameter. Finally, the fault analysis signal $f_a(k)$ is depicted. The real fault magnitude is 1, and $f_a(k)$ exhibits a high variance around the real value due to the properties of the ARX model parameters. The FDI tasks perform well, but the fault analysis does not give a good estimation of the fault magnitude. The fault occurs at time instant t_k = 240 s. The fault detection delay is 1.4 s, and the fault isolation is 5 s.

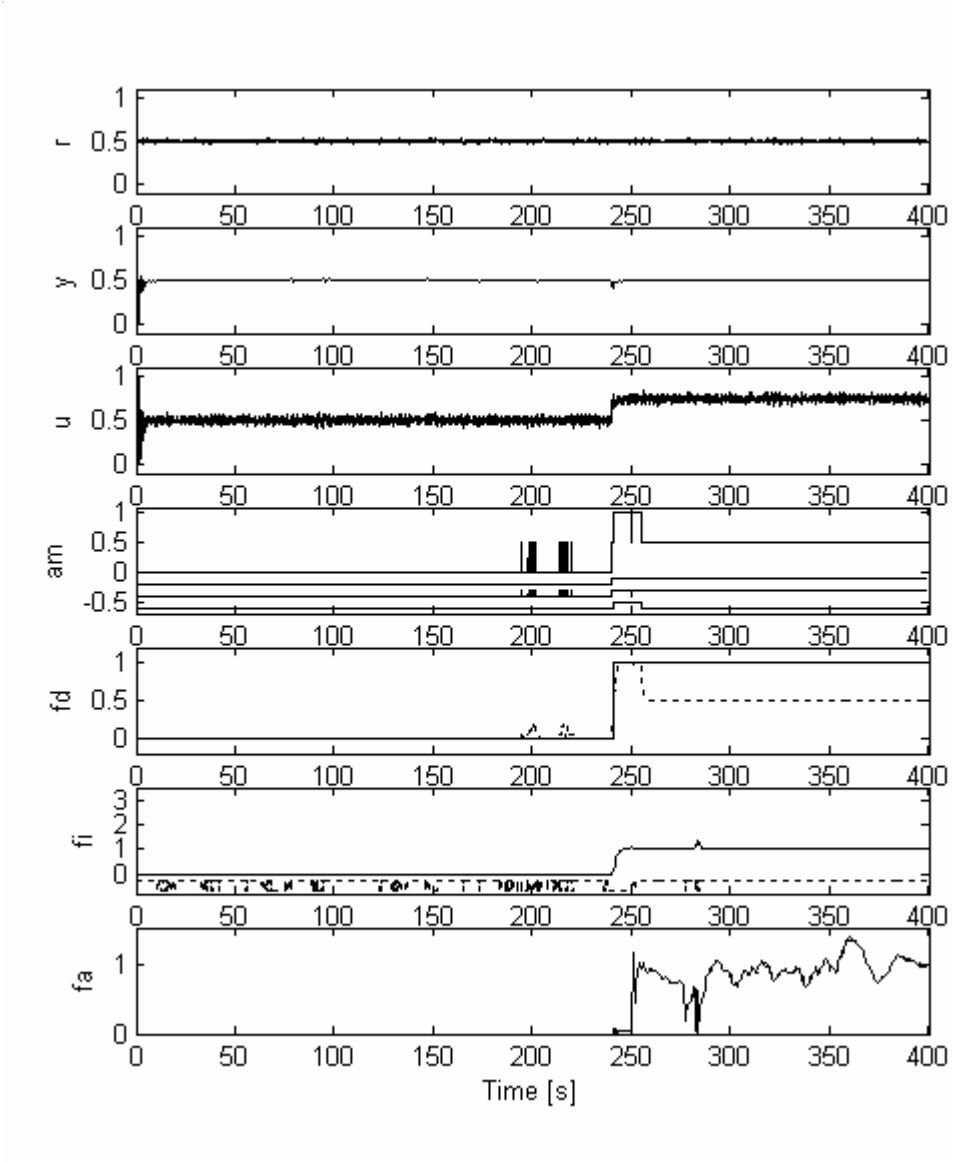


Fig. 5.23 - Input-output and FDD signals for fault F1.

The signals used for fault detection based on the PCA are shown in Fig. 5.24.

The graph at the top shows the scores space ($t_1(k)$, $t_2(k)$). The cluster associated with fault F_0 (nominal operation) is centered at the point (0; 0). The cluster associated with fault F_1 is centered approximately at the point (t_1 = -7.3; t_2 = 10.1).

The other graphs show the scores evolution for the first principal component (t1) and for the second principal component (t2), and finally the Q statistics (square of prediction error, SPE), i.e., the signal $q(k)$.

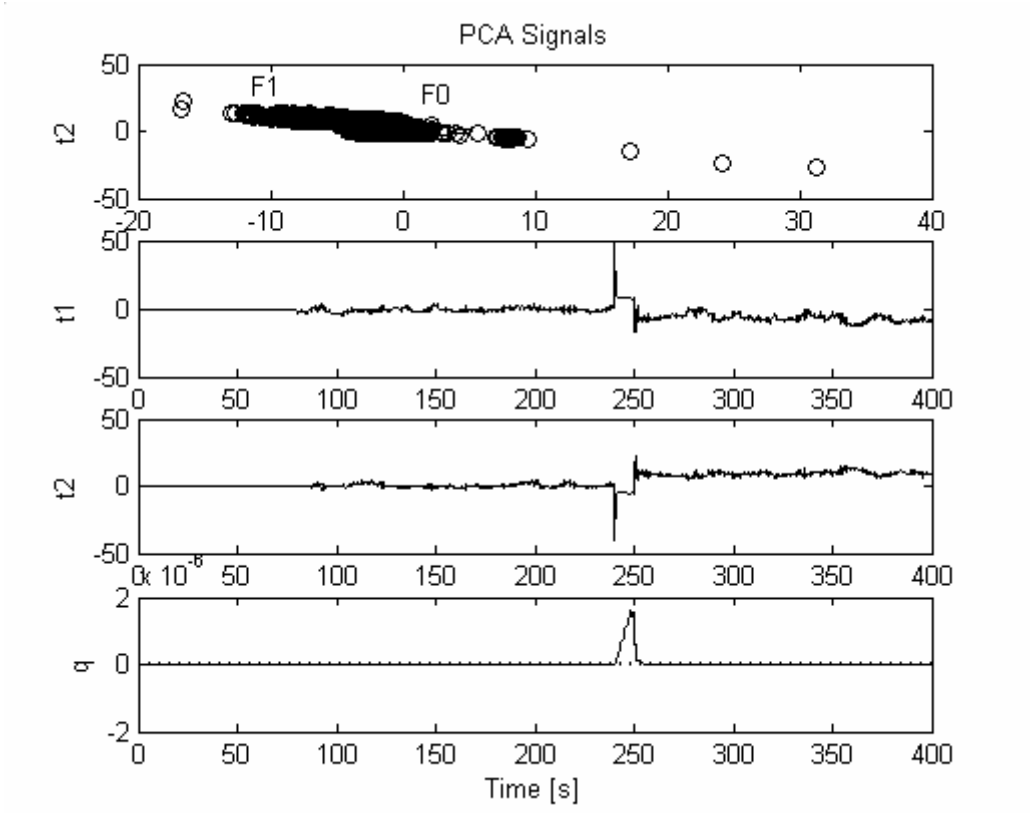


Fig. 5.24 - Signals for fault detection based on PCA.

The signals for fault diagnosis are depicted in Fig. 5.25, and Fig. 5.26.

When a fault $\Delta\gamma_i$ occurs, the feature vector deviation $\Delta\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k) - \boldsymbol{\theta}^{nom}$ will be aligned with the associated influence vector, $\boldsymbol{\Omega}_i$. In Fig. 5.25, the graphs show the angles between $\Delta\boldsymbol{\theta}(k)$ and each influence vector $\boldsymbol{\Omega}_i$, along the experiment.

The first influence vector $\boldsymbol{\Omega}_1$ is associated with the fault on the physical parameter R, and the second $\boldsymbol{\Omega}_2$ is associated with the controller proportional gain K_p. According to the minimum angle criterion for fault isolation, fault F_1 is isolated.

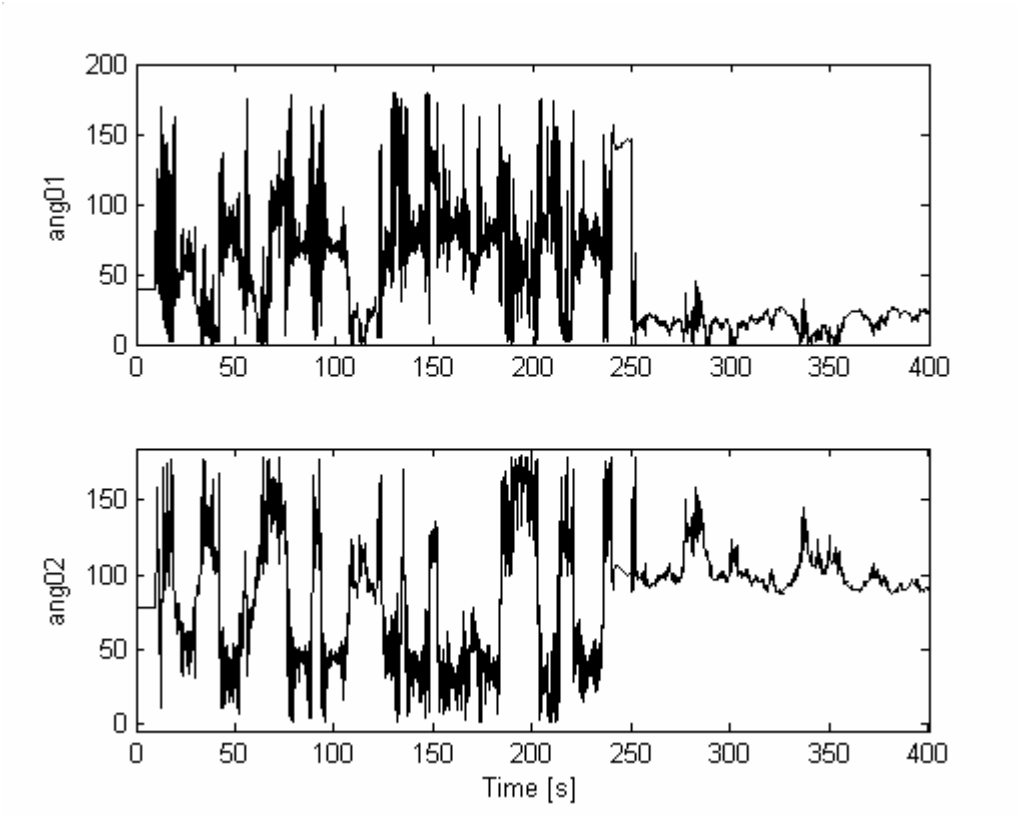


Fig. 5.25 - Signals for fault diagnosis based on IMX.

Fig. 5.26 presents a three dimensional graph, and the various projections on two-dimensional planes.

Here, a three dimensional graph is used because the ARX(2, 1, 2) model used has three parameters, [a_1 a_2 b_2]. The coordinates (p1, p2, p3) correspond to each ARX model parameter, respectively, (a_1, a_2, b_2). The feature vector deviation $\Delta\boldsymbol{\theta}(k)$ is represented by a dashed (magenta) line, and the influence vectors $\boldsymbol{\Omega}_i$ are represented in solid lines.

The first influence vector $\boldsymbol{\Omega}_1$ associated with the fault on the physical parameter R of the motor is the bigger line (red line), and the smaller line (green line) indicates the second influence vector $\boldsymbol{\Omega}_2$.

As it can be observed, $\Delta\boldsymbol{\theta}(k)$ is more aligned with $\boldsymbol{\Omega}_1$, which means that fault F_1 is isolated. Since the angle between $\Delta\boldsymbol{\theta}(k)$ and $\boldsymbol{\Omega}_1$ is less than 90º this indicates an increase of the physical parameter, as expected.

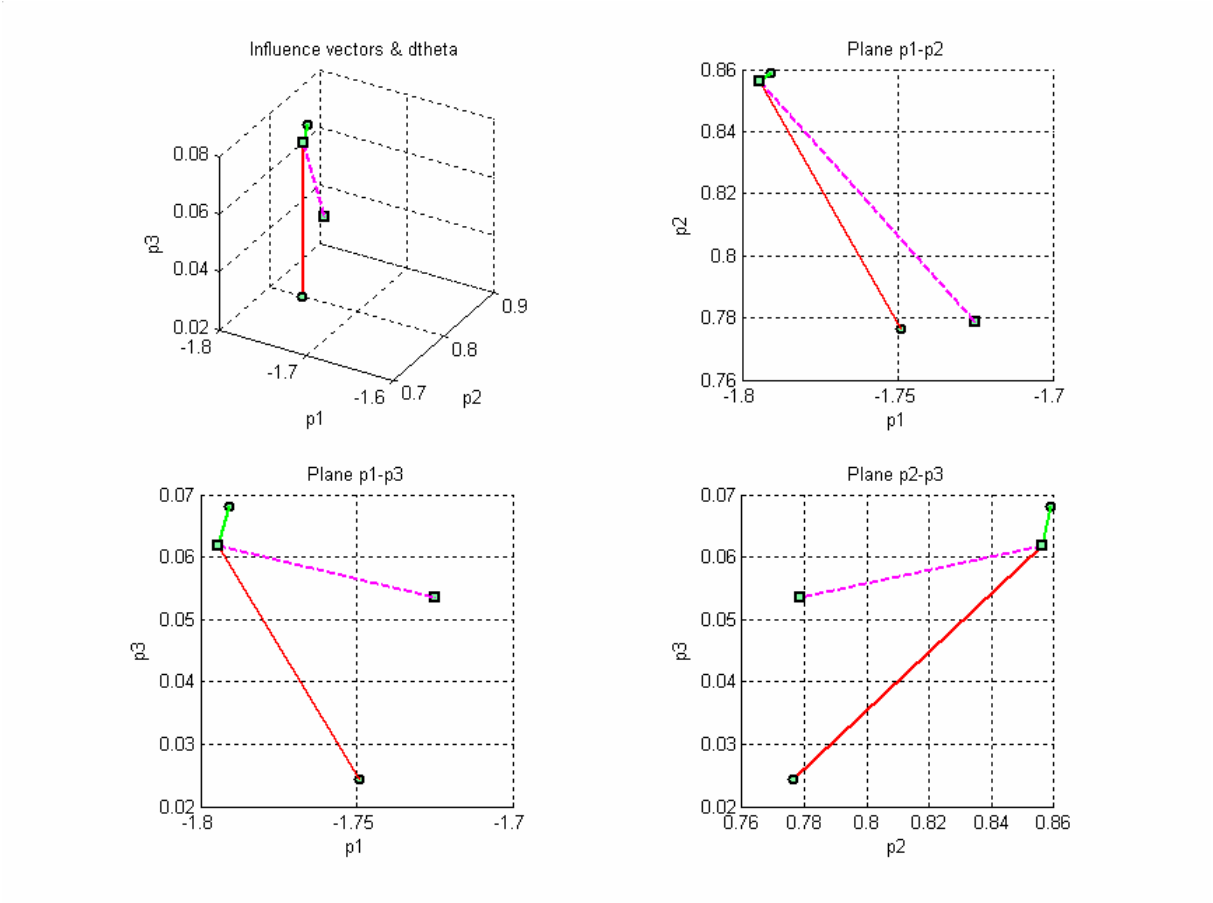


Fig. 5.26 - Feature vector deviation and influence vectors.

The second fault, F_2, is an increase in the proportional gain of the PI controller.

The gain is increased from $K_p = 2.55$ to $K_p = 25.5$, causing oscillations on input and output signals. The reference signal $r(k)$, the output signal $y(k)$ and the input signal $u(k)$ are shown in the first graphs of Fig. 5.27. The description of the signals is the same used in the previous experiment.

When the fault occurs, around $t_k = 240$ s, the system stays oscillatory, since the PI controller with fixed gains is not tuned for this faulty situation. Afterwards, the fault alarm signal $a_m(k)$ and the fault detection signal $f_d(k)$ are shown. The low pass filtering of the fault alarm signal is crucial here to avoid false detections.

The fault isolation signal $f_i(k)$ assumes the value 2, and hence fault F_2 is isolated. Finally, the fault analysis signal $f_a(k)$ indicates an estimation of the fault magnitude around 46, a value approximately the double of the real fault magnitude. This estimation error is due to the nonlinear relations that exist between ARX model parameters and physical parameters.

The fault occurs at time instant $t_k = 240$ s, the fault detection delay is 1.2 s, and the fault isolation delay is 5.5 s.

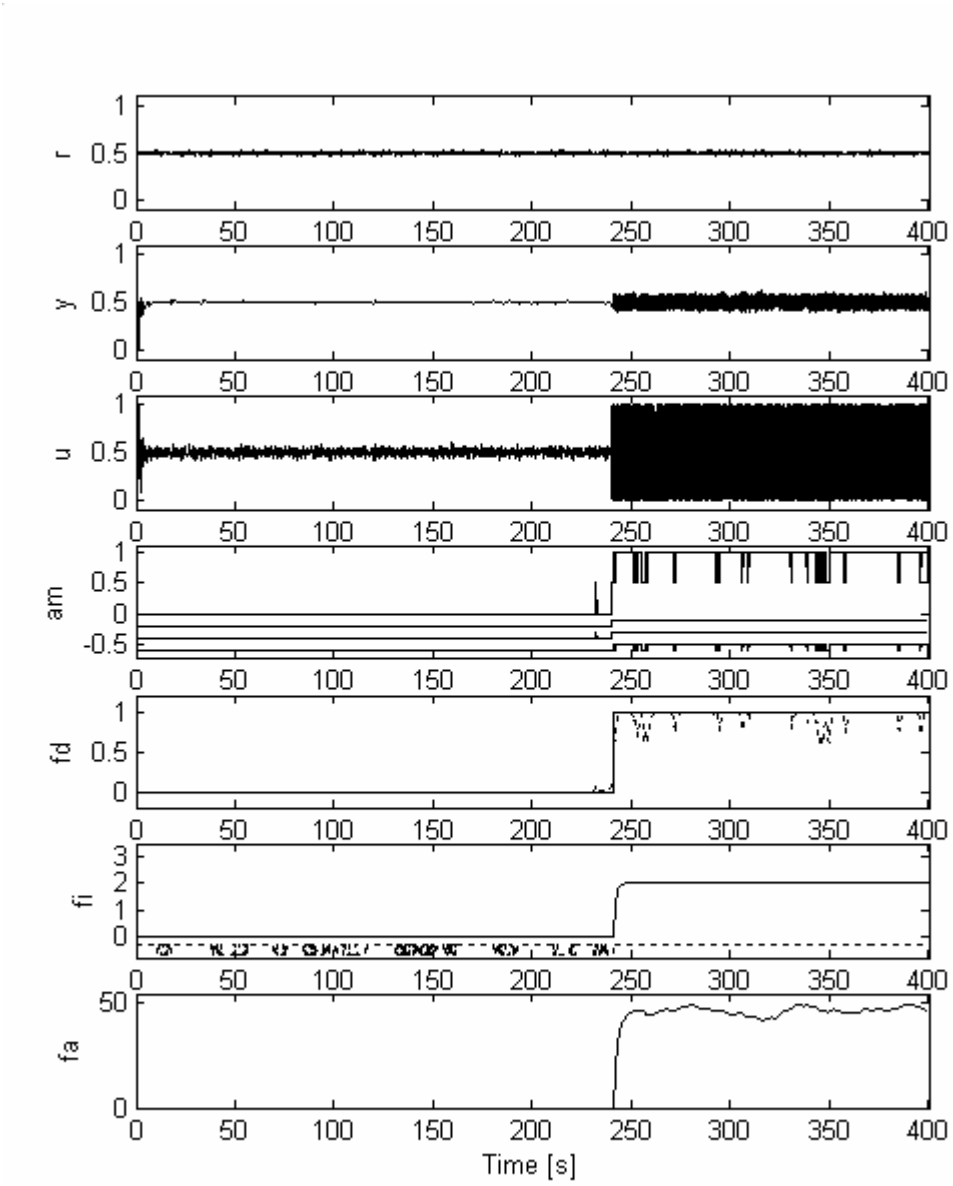


Fig. 5.27 - Input-output and FDD signals for fault F2.

The system exhibits oscillations after fault occurrence, and for this case the ARX model parameters present a small variance compared to the case of fault F_1. Indirectly, this is reflected in the angles between the feature vector deviation $\Delta\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k) - \boldsymbol{\theta}^{nom}$ and the influence vectors $\boldsymbol{\Omega}_i$ as depicted in Fig. 5.28.

A positive angle (label "ang02") near zero indicates a positive fault magnitude on fault F_2, as shown in the magnitude graph (signal $f_a(k)$ in Fig. 5.27).

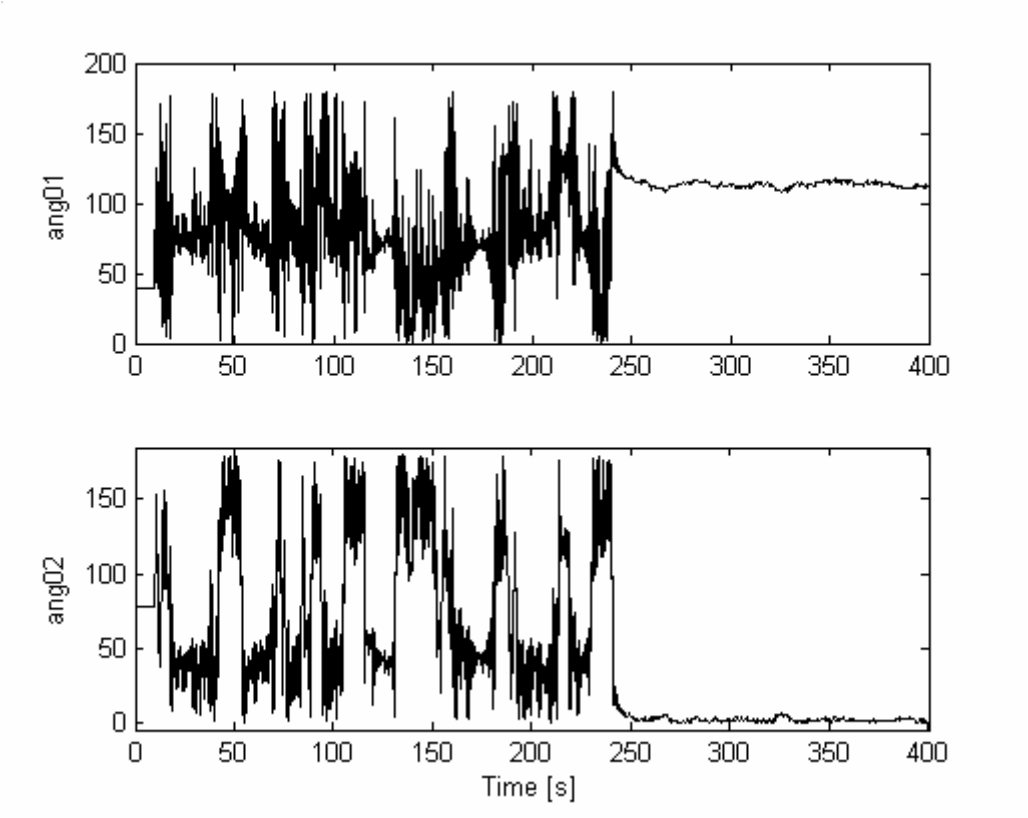


Fig. 5.28 - Signals for fault diagnosis based on IMX.

In the experiments shown before, in this section, the process (plant) works in steady-state around a set-point. In the next experiment some variations on the set-point are done, and the effect on the FDD performance is described. For the fault detection and diagnosis approaches based on on-line parameter identification, it is expected that the set-point variations be understood as temporary faults. After a set-point variation new input-output is available, and the sliding window parameter estimation algorithm (SW-PCR) needs a certain time (equivalent to its window length) to adapt to the new operating conditions. In fact, this is what is observed in the experiment.

Fig. 5.29 depicts the reference (set-point) signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. Next, the parameters of the ARX reference-output model M_{yr}, that relates the output signal $y(k)$ and the reference signal $r(k)$ appear. Finally, the fault detection signal $f_d(k)$ appears where the temporary faults are observed. The duration of each pulse on the fault detection signal is approximately given by {11; 13; 12; 14} s. This duration is a function of the window length (10 s, here) of the SW-PCR parameter estimation algorithm, and also depends on the delay introduced by the low pass filtering of the fault alarm signal.

One possible solution to avoid this drawback is to temporarily deactivate the fault detection task during a change on the set-point.

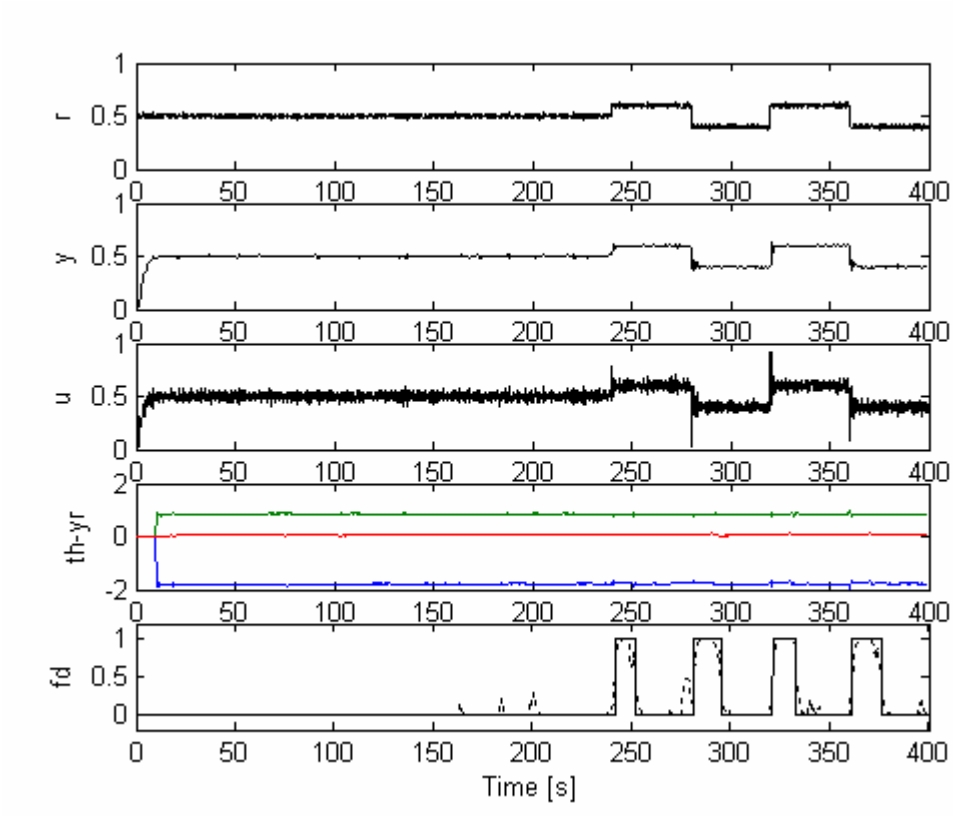


Fig. 5.29 - Set-point variations and transient effects.

Remarks.

For the parametric faults evaluated, the overall performance of the proposed FDD methodology is reasonable. Both tasks of fault detection and fault diagnosis require the on-line estimation of ARX model parameters, and hence good persistent excitation conditions are required. This FDD approach can be applied to SISO or MIMO systems.

For the fault detection approach based on PCA applied to the ARX model parameters, the feature vector is based on the scores and on the SPE (*Q* statistics). The scores are monitored using the *T2* statistics, and a three sigma limit is used to monitor the SPE. The data does not obey a perfectly normal distribution, but the normality assumption used for definition of thresholds gives a reasonable performance in terms of false alarms. In practical situations, sometimes the thresholds must be adjusted to guarantee a better performance. The low pass filtering of signals also plays an important role, avoiding wrong fault detections and isolations.

The performance of the fault diagnosis approach based on the influence matrix method depends strongly on the multi-linearity between model parameters and physical process parameters. In practice, for small deviations on the physical parameters this multi-linearity is verified. For high deviations, the relations deviate from the multi-linearity and cause a degradation of the performance, rendering the diagnosis of small faults difficult.

For nonlinear systems, it is also possible to apply this FDD combined method using a multi-model approach. A set of PCA and IMX models must be obtained for each set-point. The implementation of this multi-model approach requires a supervisor system to manage a switching strategy between models.

5.9 FDD Approach for Nonlinear Systems based on NROP Predictors

For nonlinear systems, a new fault detection and diagnosis approach was proposed in section 4.4. The FDD architecture proposed is here again depicted in Fig. 5.30. This FDD approach is based on a bank of neural recurrent output predictors (NROP). One NROP is tuned to the nominal operating region, and the others are tuned to each faulty situation. A fault alarm $a_m(k)$ is generated if the square of prediction error $q(k)$ exceeds the threshold values, **h**. A fault is detected by low pass filtering $a_m(k)$ and by thresholding. Fault isolation is executed based on the best tuned NROP predictor, i.e., the one with the smallest square of prediction error, $q_j(k)$.

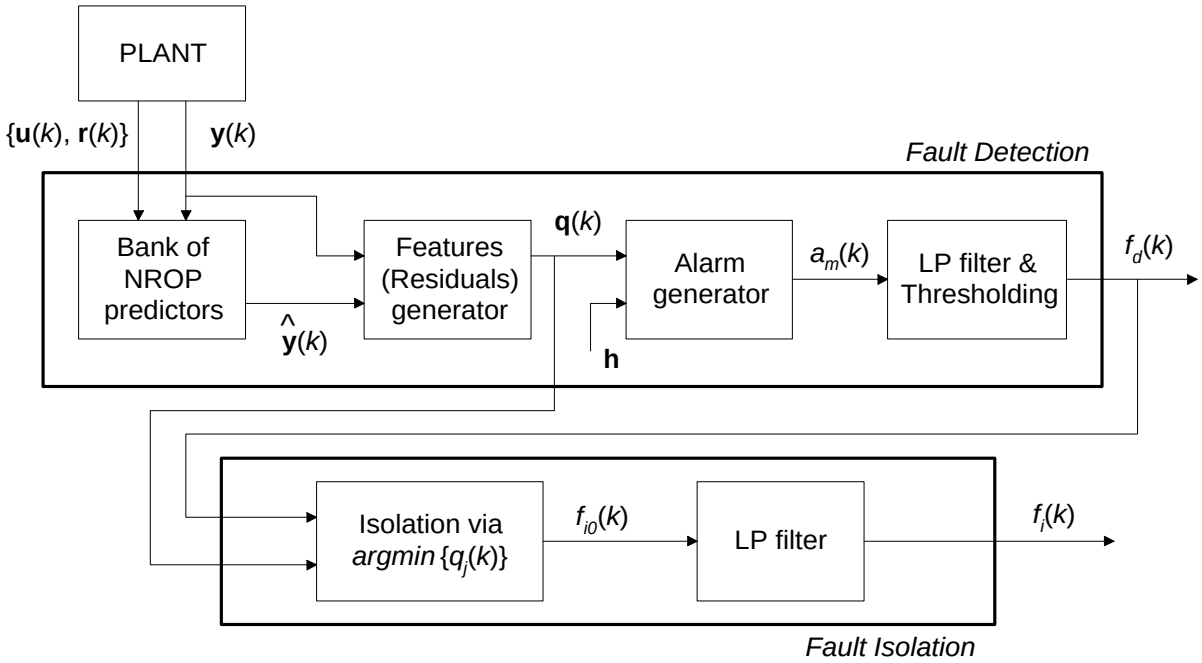


Fig. 5.30 - Architecture of the FDD approach based on a bank of NROP predictors.

Experimental results obtained with the nonlinear DC motor (DCM-RA) setup are presented later. Four neural predictor models have been trained off-line for each faulty situation. Each NROP predictor $\Psi_i(\mathbf{W}_i,\ldots)$, $i = 0, \ldots, 3$, has an embedded neural predictor model. In this experiment, neural nonlinear NARX(3, 1, 2) models are used to model the process. The nonlinear NARX neural process models relate the output $y(k)$ and the input $u(k)$ signals. After the off-line training phase, the output predictor models are incorporated in the NROP architecture (Fig. 4.3), in order to be used on-line for fault detection and isolation. A gain of $K_n = 0.1$ was selected for each NROP neural predictor (Eq. 4.7).

This FDD method based on a bank of NROP predictors is applied to a nonlinear DC motor setup (DCM-RA). The setup is described in detail in section 5.3.3. The operating conditions used in this experiment are the following. An adaptive optimal linear quadratic Gaussian controller (LQGC) is used in this experiment to control the speed of the DC motor setup, using a design parameter $r_0 = 0.4$, and low pass filtering the LQGC gains by an IIR filter $H_{lp}(z, \lambda)$ with pole located at $\lambda = 0.9$. At the start-up, a PI controller is used, with parameters $K_p = 1$, and $T_i = 2$ s. A dither with variance 1×10^{-3} has been added to the reference signal.

The set of faults is denoted by $F = \{F_0, F_1, F_2, F_3\}$. The nominal operating region, assumed for set-points greater than 0.5, is denoted by fault F_0. Here, a set-point of 0.7 is used for tests. Fault F_1 corresponds to a decrease of the actuator gain in 50%. Fault F_2 corresponds to a blocked sensor at value 0.6, making the plant unobservable. Fault F_3 corresponds to a change to a critical operating region (a set-point around 0.4) outside the nominal region, where the speed of the motor is near a minimum acceptable value.
At the startup a PI controller is used, and at time instant $t_k = 80$ s the supervisor switches to the adaptive optimal LQG controller based on an ARX(2, 1, 2) model identified on-line.

In the first experiment, the first two faults (F_1 and F_2) occur at different time instants, as it can be observed in Fig. 5.31 looking at their effect on the output signal $y(k)$. The reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$ can be observed. Next the fault alarm signal $a_m(k)$, the fault detection signal $f_d(k)$, and the fault isolation signal $f_i(k)$ appear. The fault magnitude is not considered here, and that is the reason $f_a(k)$ assumes the zero value.

Fault F_1 occurs at time instant $t_k = 240$ s, its duration is 40 s, the detection delay is 2.1 s, and the isolation delay is 10 s. For the fault F_2, it occurs at $t_k = 320$ s and continues until the end of the experiment. The detection delay is 2.3 s, and the fault isolation delay is 11 s. Both faults cause a saturation of the actuator.

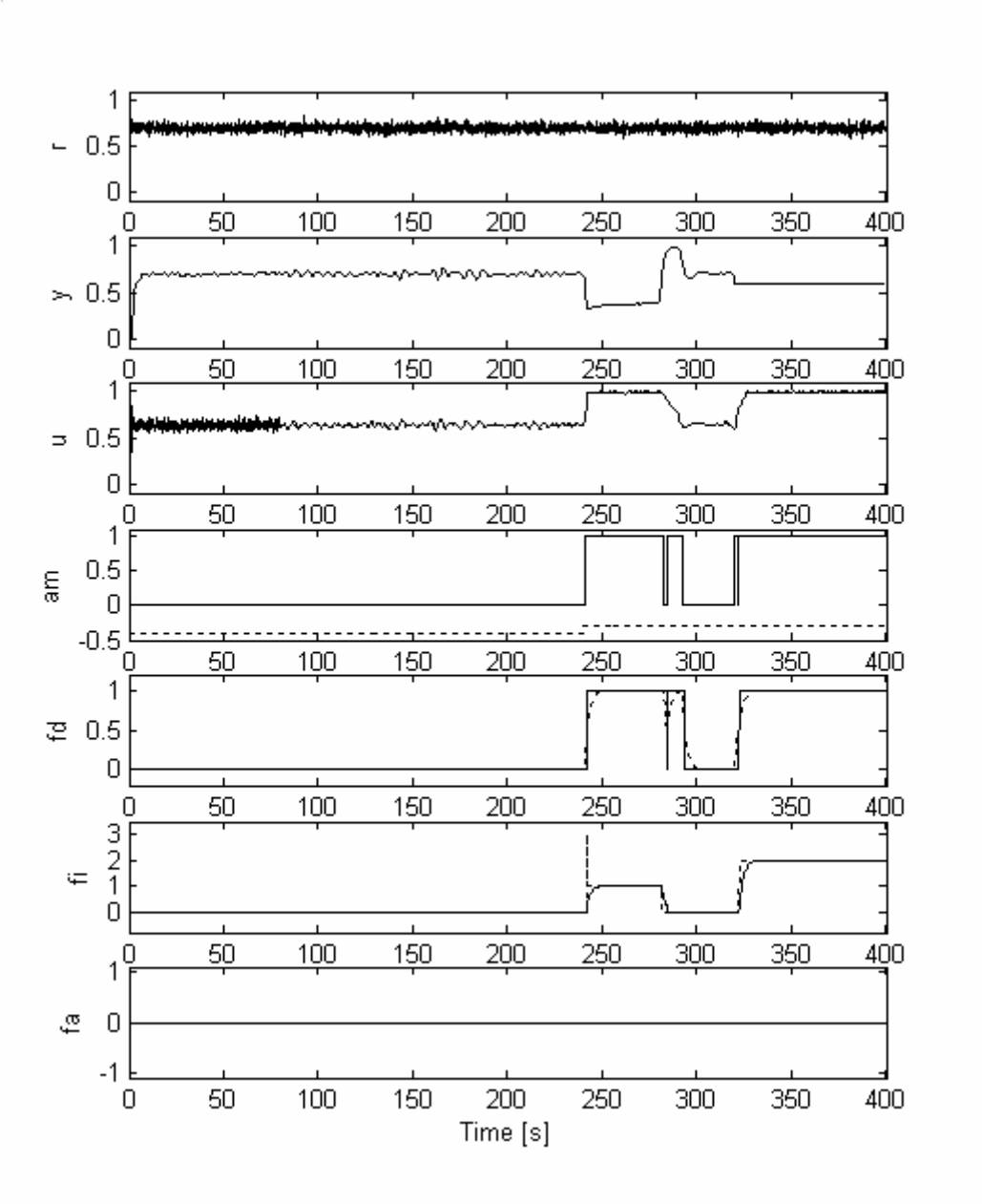


Fig. 5.31 - NROP approach. FDD signals for faults F1 and F2.

Fig. 5.32 shows the square of prediction errors (SPE's) computed on-line based on the NROP output residuals. These signals are used for fault detection, and for fault isolation as described in the proposed approach (section 4.4). The SPE signal for the NROP tuned to fault F_1 is indicated with the corresponding label F1. As expected, when fault F_1 occurs (between 240 s and 280 s), the SPE signal (red line) for the nominal region (fault F_0) exceeds the thresholds,

and the SPE signal (green line) for the NROP tuned to fault F_1 goes to a value around zero. The isolation is performed according to the smallest SPE signal criterion. It is interesting to observe the output oscillations occuring on some NROP neural predictors, when no faults occur. This happens because the NROP predictors have been trained off-line based on specific faulty data, and each one has an embedded faulty model. Each NROP predictor $\Psi_i(\mathbf{W}_i,\ldots)$ converges well when the associated fault F_i occurs. This occurs if the embedded neural model has been trained with richness data for the faulty case. The fault signal F_2 is represented by the blue line, and the fault signal F_3 by the magenta line.

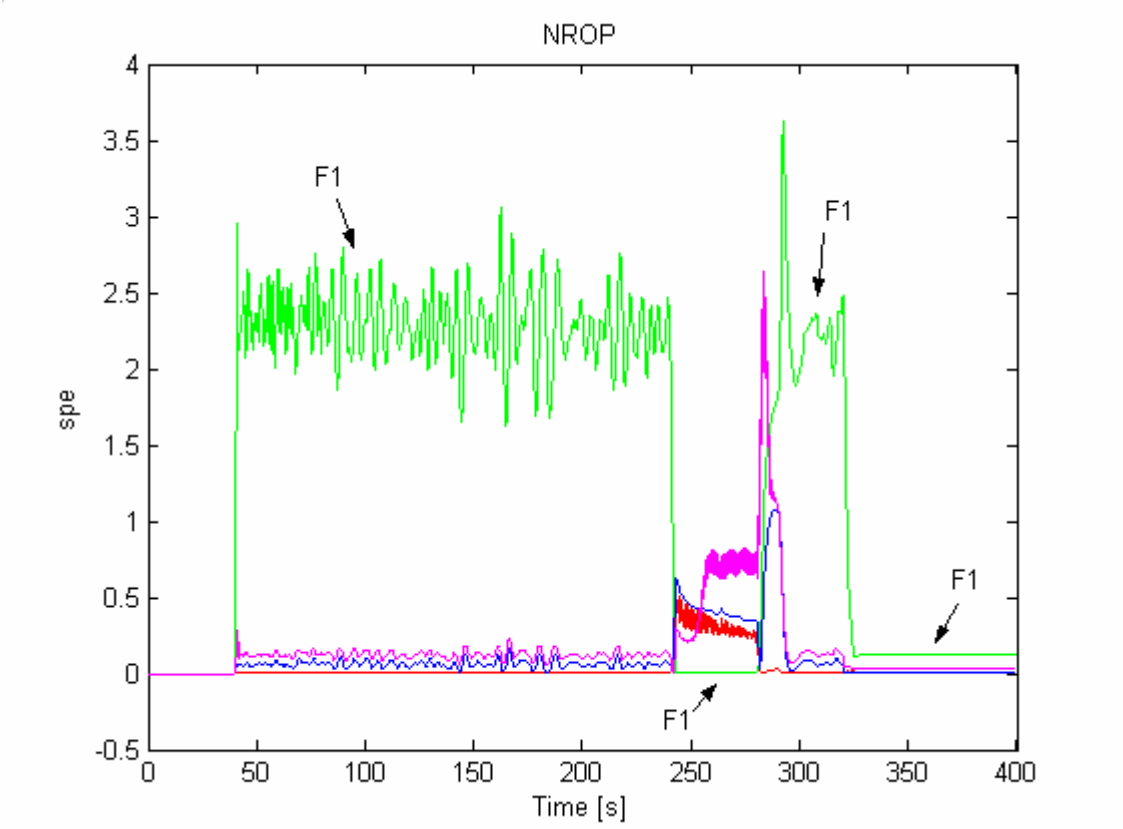


Fig. 5.32 - SPE of output residuals for fault detection and isolation.

In Fig. 5.33, some signals can be observed for faulty situation F_3. The system changes from the set-point 0.7 to 0.4, and this new set-point here is considered a critical operating region (a faulty situation), since the motor speed is near the minimum admissible value. For lower set-points the system cannot work, because the applied voltage is not sufficient to guarantee the rotation of the DC motor. When the fault occurs, during a small time interval, the real motor speed goes to zero. In Fig. 5.33, the first three graphs show, respectively, the reference (set-point) signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. Next the fault alarm signal $a_m(k)$ can be observed. The fault detection signal $f_d(k)$ appears next, and the detection delay is

2.5 s. The isolation delay is 12 s, as depicted in the signal $f_i(k)$. The fault analysis signal $f_a(k)$ is zero, since it is not considered here.

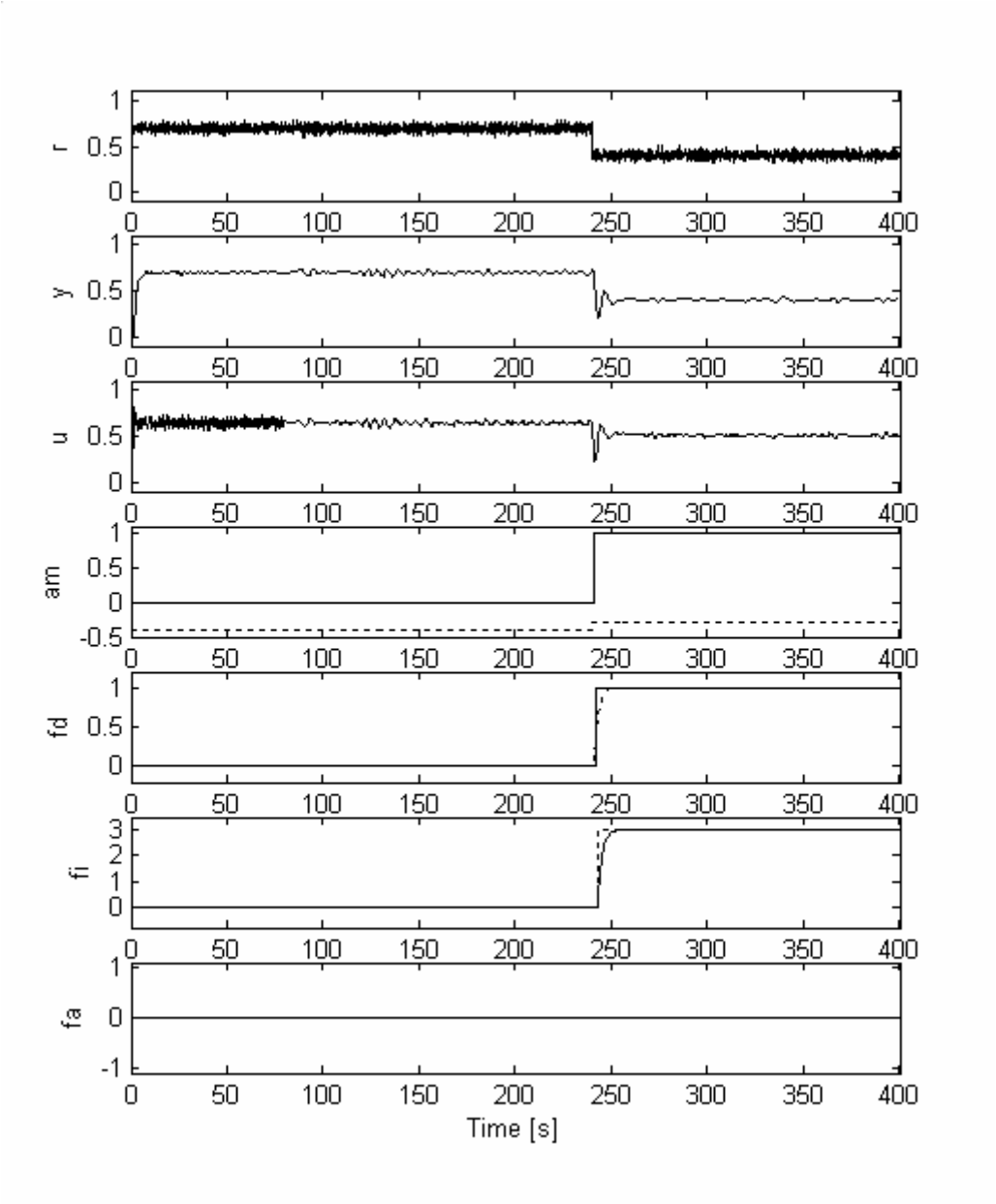


Fig. 5.33 - NROP approach. FDD signals for fault F3.

In the first graph of Fig. 5.34 the reference signal (red line) and the input-output signals (green line and blue line, respectively) are shown. The other graphs show the output process signal $y(k)$ (in dotted line) and each output NROP predictor signal $\hat{y}_{nrop}(k)$ for each faulty situation in solid line. The second graph shows that, in nominal operation, the output predictor signal for the NROP tuned to fault F_0 is approximately equal to the process output signal, as expected.

After fault occurrence, for the NROP tuned to nominal operation (fault F_0), the output predictor error $y(k) - \hat{y}_{nrop}(k)$ increases, and the fault is detected since the SPE exceeds the threshold.

After fault occurrence, the best tuned NROP predictor is the one shown in the last graph, i.e., the NROP tuned to fault F_3.

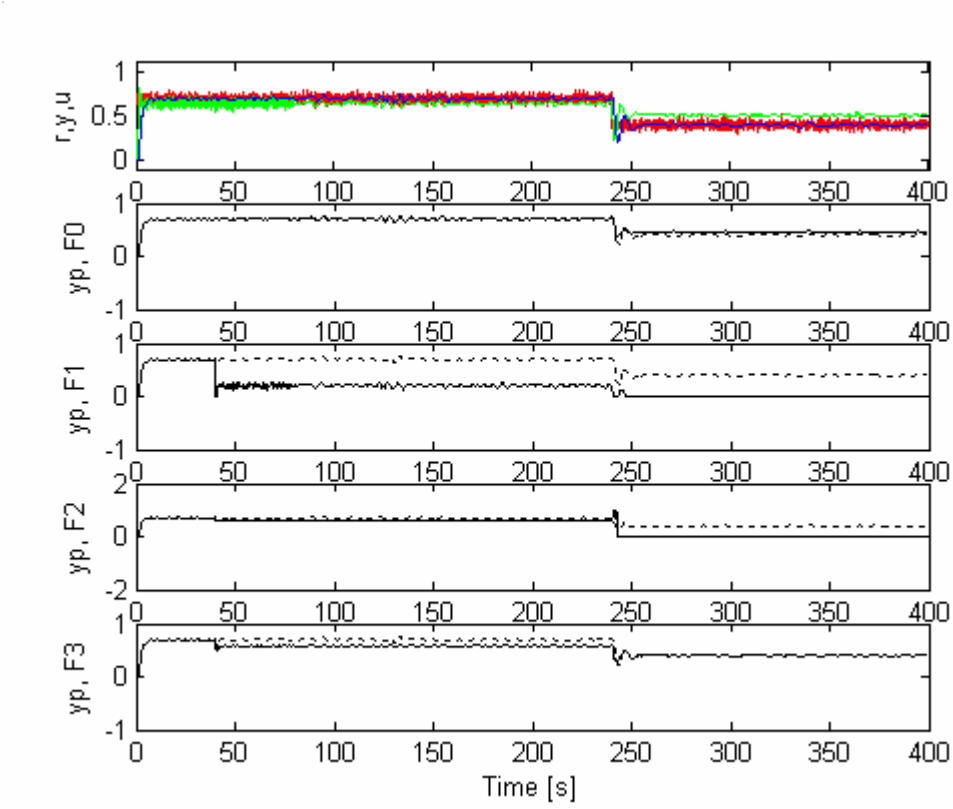


Fig. 5.34 - Output predictor signals for NROP approach.

The gain K_n of the neural NROP predictor is a design parameter, allowing the engineer to adjust the predictor performance, and consequently the FDD performance. Some experiments have been performed with the DC motor setup (DCM-RA), varying the gain K_n for the NROP predictor tuned to the nominal operation (fault F_0).

Tab. 5.3 shows the mean values obtained for the square of prediction error (SPE), $q(k)$, for different gains. A decrease of the SPE error can be seen and then an increase. There is a range of gains with low acceptable SPE errors; a value in this range must be chosen. The value chosen for the gain of the NROP should guarantee a stable behaviour of the output predictor, and also good fault isolation properties.

Tab. 5.3 - Square of prediction error (SPE) versus gain for NROP predictor.

Gain, K_n	Mean($q(k)$)
0	4.3×10^{-2}
0.05	1.9×10^{-3}
0.1	5.5×10^{-4}
1	9.6×10^{-6}
2	3.3×10^{-6}
5	2.5
10	2.5

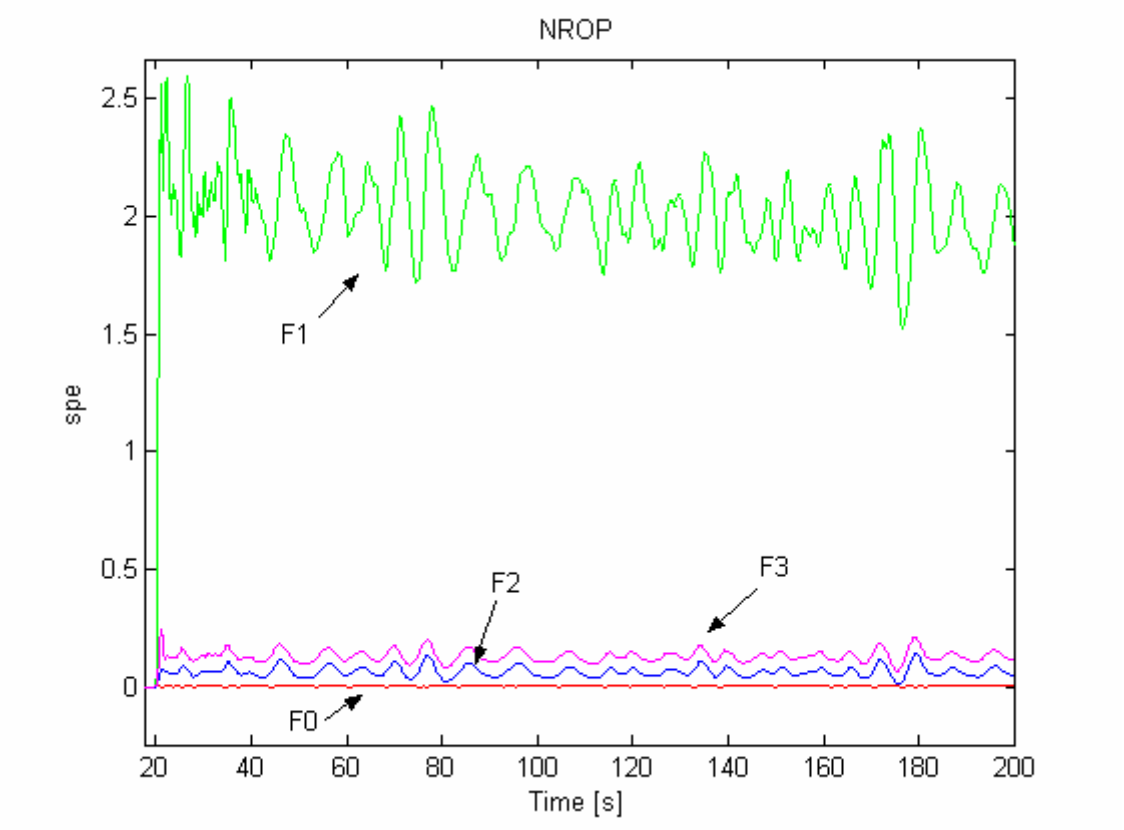


Fig. 5.35 - SPE of residual signals for NROP predictors.

An experiment was performed with the DC motor setup (DCM-RA) for 200 s. The SPE for each NROP predictor with gain $K_n = 0.1$, are depicted in Fig. 5.35, for the case of nominal operation (fault F_0). For the FDD approach proposed in this work, the fault isolation criterion is based on the NROP predictor with the smallest SPE signal. As expected in this experiment

for nominal operation, the NROP tuned to fault F_0 is the one with the smallest SPE signal, with a mean value of 5.5×10^{-4} as observed in Tab. 5.3.

Remarks.

The results obtained with this new fault detection and diagnosis approach based on neural recurrent output predictors (NROP) show a good performance for the set of faults considered. This approach is appropriate to detect different kinds of faults: additive faults, some parametric faults, faults that cause strong output oscillations, and some kinds of structural faults like blocked sensors, saturations, etc. The great potential of this FDD approach is the fact that it is based on neural network predictor models, and these neural models are able to captured nonlinear behaviours.

The performance of this FDD approach is heavily dependent on the quality of the embedded neural output predictor models. For each faulty situation, it is necessary to identify a neural output predictor model off-line, by training a FF-MLP neural network with informative data. The NROP predictor performance can be adjusted by selecting different gain values, K_n. This design parameter, K_n, allows to settle the magnitude of the output prediction error, and consequently the fault detection and isolation performance.

5.10 FDD Based on Neural NLPCA and Neural NNLDA

In this section some results are presented which were obtained for the combined fault detection and diagnosis (FDD) approach based on neural nonlinear principal component analysis (NLPCA) and on neural nonlinear discriminant analysis (NNLDA). This nonlinear FDD approach, described in detail in section 4.5.5, obeys the architecture depicted in Fig. 5.36.

On-line estimation of ARX model parameters is performed based on the SW-PCR algorithm. The NLPCA approach is applied to the ARX model M_{yr} parameters estimated on-line, generating the features for FDD. The features are the two dimensional scores $\mathbf{t}_a(k)$ and the square of prediction error $q(k)$. A different feature (pattern vector) is associated to each fault. This pattern is classified by a neural NNLDA discriminant approach, and a class is attributed. A fault alarm $a_m(k)$ is generated if the on-line pattern is outside the nominal region. The fault detection signal $f_d(k)$ is obtained by low pass filtering $a_m(k)$, and by thresholding. The fault isolation is achieved by fault classification and low pass filtering.

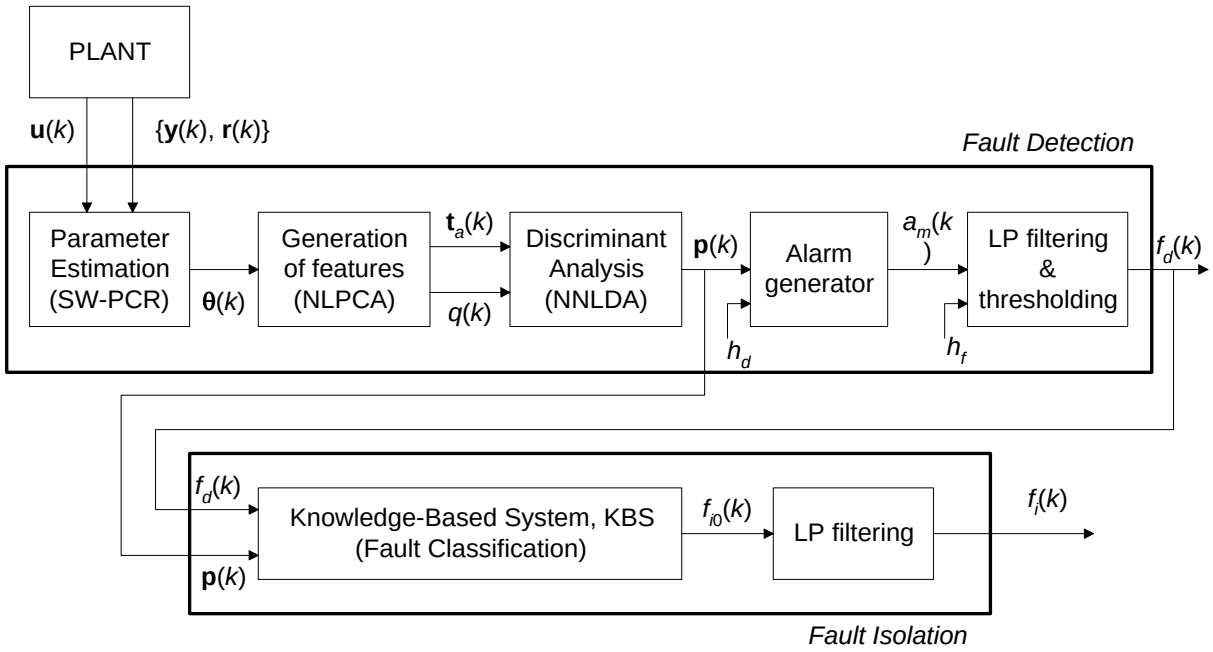


Fig. 5.36 - Architecture of FDD approach based on neural NLPCA and neural NNLDA.

The experiments shown here were performed with the nonlinear DC motor setup (DCM-RA) described in detail in section 5.3.3. The input data has been auto-scaled, removing the mean and divided by the standard deviation of the nominal data used for building the NLPCA nominal model. The input data are the ARX model parameters. Auto-scaling standardizes the process variables, ensuring that for each variable an equal weight is given; this means, for this case, that both a_i and b_j parameters have the same weight.

An ARX(2, 1, 2) model M_{yr}, relating the output signal $y(k)$ and the reference signal $r(k)$, has been considered in this experiment for modeling the closed-loop system dynamics. This model enables the detection and diagnosis of faults on the process and also on the controller.

A neural network has been trained for the NLPCA, with nominal data captured around the set point 0.7, during 200 epochs using 3636 samples, and a SSE error of 1.0×10^{-2} was achieved. The explained variance by the NLPCA model for the nominal training data, computed according to Eq. 4.23, is greater than 99 %.

The ARX model M_{yr} presents a smaller variance on the parameters than the parameters of the model M_{yu}, since the dither signal is added to the reference signal and the identification of M_{yr} can be considered in open-loop. The SW-PCR parameter estimation algorithm is used for on-line estimation of the ARX parameters, using a window length of 10 s.

The faults considered in this experiment are the nominal operation termed fault F_0 and three more faults. The parametric fault F_1 corresponds to the removal of the digital filter $H_{lp}(z, \lambda)$ from the process output, putting a zero value on the filter design parameter, $\lambda = 0$, and

consequently turning the system faster. A sensor blocked at value 0.5 is designated fault F_2, a structural fault. The parametric fault F_3 is a change on the design parameter r_0 of the LQG controller from 0.4 to 0, causing output oscillations. Each fault corresponds to a different set of ARX model parameters, and consequently a different set of scores and SPE signal, as shown in Fig. 5.37.

The neural network that implements the neural discriminant analysis (NNLDA) has input patterns given by the scores and the SPE signal, both shown in Fig. 5.37. For this neural network, after 500 epochs of training and using 14400 samples, a SSE error of 5.1×10^{-4} was reached.

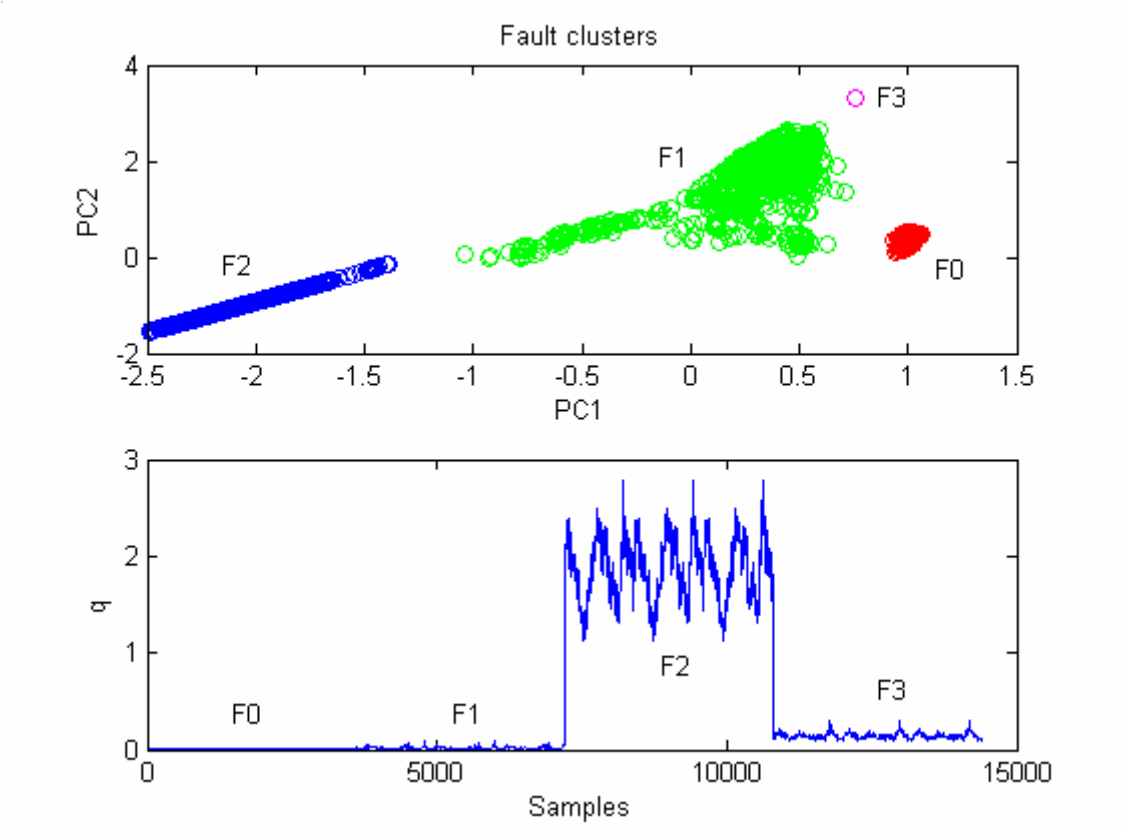


Fig. 5.37 - Fault clusters in scores space, and SPE signal.

In nominal operation, the input data for the neural network used for NLPCA are the three ARX parameters $\{a_1(k), a_2(k), b_2(k)\}$, and the output data are the two dimensional scores $\{t_1(k), t_2(k)\}$ and the SPE signal $q(k)$ of the residuals.

The nominal operating point is around 0.7, and the variance of the dither signal added to the reference signal is 1×10^{-3}. From start-up and until 80 s the classical PI controller is the active controller, using gains $K_p = 1$ and $T_i = 2$ s. After, the supervisor switches to the adaptive optimal LQG controller. The LQG controller parameters are computed on-line based on a

linearized ARX(2, 1, 2) model M_{yu}. A value of 0.4 is used for the design parameter r_0 of the LQG controller, in order to adjust the closed-loop dynamics. The nominal parameters of ARX(2, 1, 2) model M_{yu} are given by $[a_1\ a_2\ b_2] = [-1.86\ 0.87\ 1.57\times10^{-2}]$. For the ARX(2, 1, 2) model M_{yr}, the nominal vector is given by $[a_1\ a_2\ b_2] = [-1.83\ 0.90\ 6.74\times10^{-2}]$.

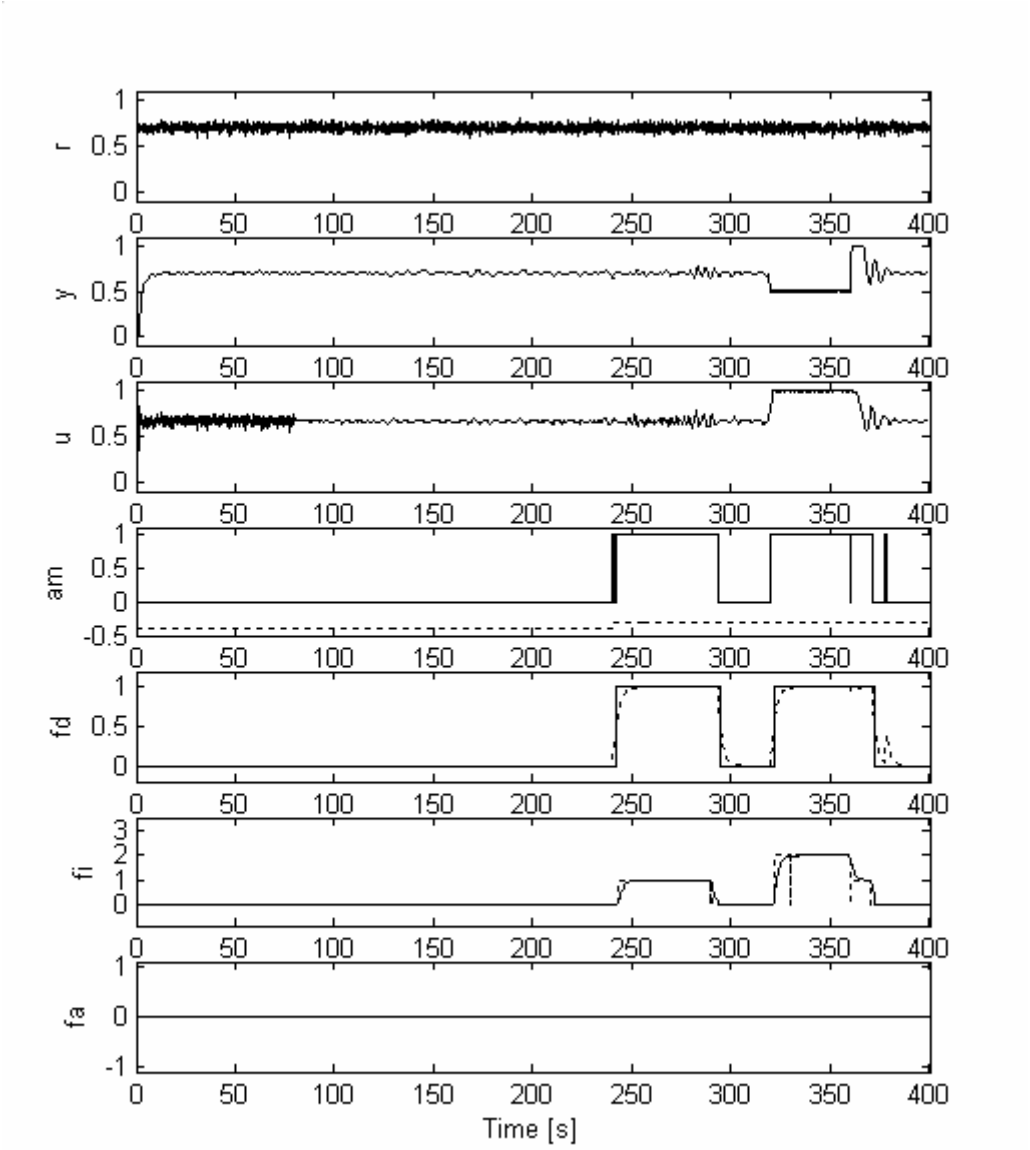


Fig. 5.38 - Input-output and FDD signals for faults F1 and F2.

In the first experiment, only the faults F_1 and F_2 appear, occurring at time instants $t_k = 240$ s and $t_k = 320$ s, for 40 s each. The results are depicted in Fig. 5.38, which shows the reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$. Next, the fault alarm signal $a_m(k)$, the fault detection signal $f_d(k)$, and the fault isolation signal $f_i(k)$ appear. The fault magnitude is not considered here, and that is the reason $f_a(k)$ assumes the zero value.

In the faulty situation F_1 the system becomes faster, since the filter $H_{lp}(z, \lambda)$ is removed by setting the filter parameter, $\lambda = 0$. The detection delay is 2.3 s, and the isolation delay is 10 s. By only observing the input and output signals, this fault is difficult to detect and diagnose.

For the fault F_2 the sensor stays blocked at value 0.5, a positive control error occurs and consequently the actuator saturates. The detection delay is 1.5 s, and the isolation delay is 10 s. The isolation delay depends on the length of the sliding window (10 s) and on the low pass filter design parameter.

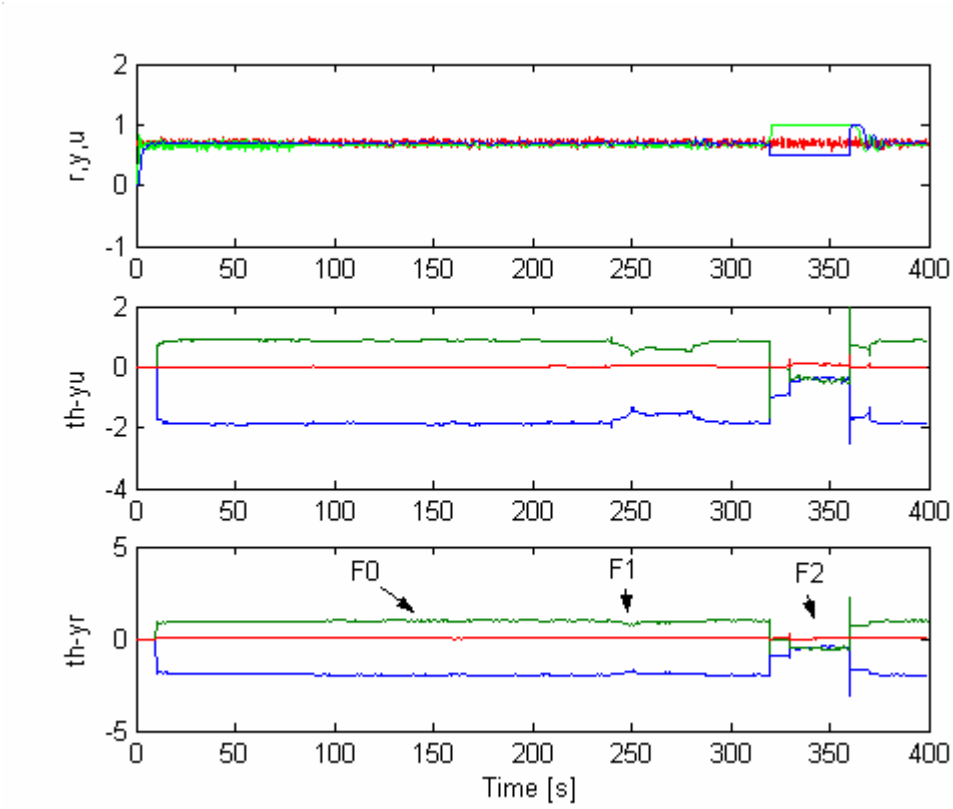


Fig. 5.39 - Input-output signals and ARX model parameters for faults F1 and F2.

Fig. 5.39 shows, in the top graph, the reference signal $r(k)$ (red line), the output signal $y(k)$ (blue line) and the input signal $u(k)$ (green line). The middle graph shows the parameters of the ARX model M_{yu}, not used in this experiment. In the last graph, the ARX model (M_{yr}) parameters used in this experiment are depicted, and the different faulty regions are indicated. When the fault F_2 occurs can be observed the high derivative on the ARX model parameters, and also the change of the model parameters after the length (10 s) of the sliding window (SW-PCR) parameter estimation algorithm.

Some experimental results are shown next for the faulty case F_3. This fault is an abrupt change of the design parameter r_0 of the LQG controller, from 0.4 to 0. One of the symptoms of this fault is the high variance of the control input signal $u(k)$, as observed in Fig. 5.40.

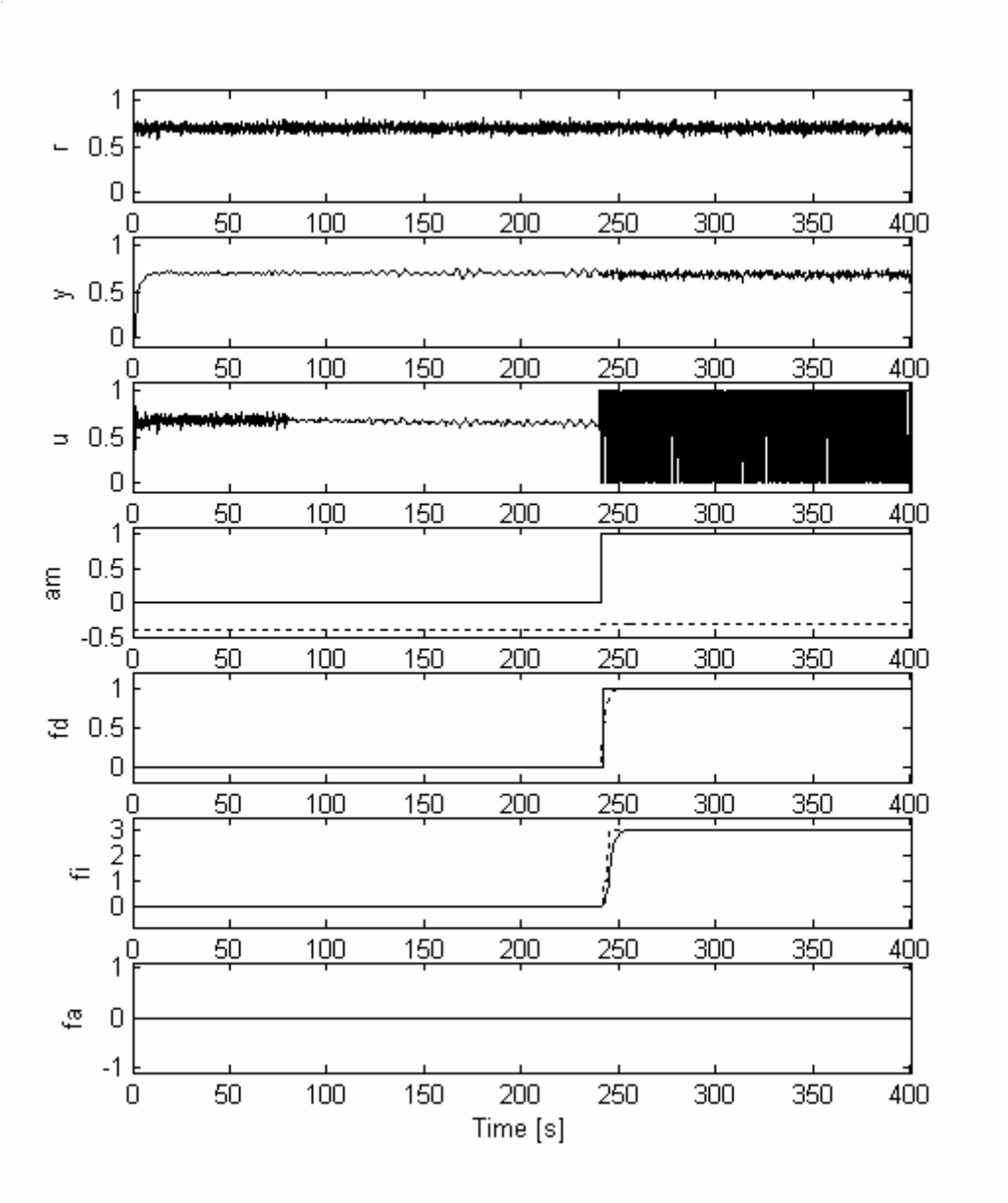


Fig. 5.40 - Input-output and FDD signals for fault F3.

Input-output and FDD signals can be observed in Fig. 5.40. From top to bottom, the reference signal $r(k)$, the output signal $y(k)$, and the input signal $u(k)$ are depicted. Next, the fault alarm signal $a_m(k)$, the fault detection signal $f_d(k)$, and the fault isolation signal $f_i(k)$ appear. The fault analysis is not considered here, and that is the reason why the signal $f_a(k)$ is zero.

The fault detection delay is 2.3 s, and the isolation takes place 14 s after fault occurrence. The main visual symptom of this fault is the strong oscillations on the system input.

The variation of the real speed motor is also significant, but here this is not perceptible due to the various low pass filters embedded into the DCM-RA motor setup. The slow dynamics is due to the frequency-to-voltage converter, the anti-aliasing filter, and the digital low pass filter.

In Fig. 5.41, in the first graph the input data **X** (ARX model parameters, in green lines) can be observed, and also the estimated values $\mathbf{X}_e$ (blue lines) by the NLPCA model. Next, the SPE error appears, $q(k)$, computed from the NLPCA residual. The next graph shows the evolution of the scores (PC1 in red line and PC2 in green line) as a function of the time. Finally, the last graph shows the two dimensional scores space (associated with the principal components PC1 and PC2) where the square symbols show the location of each fault cluster (pattern) centre.

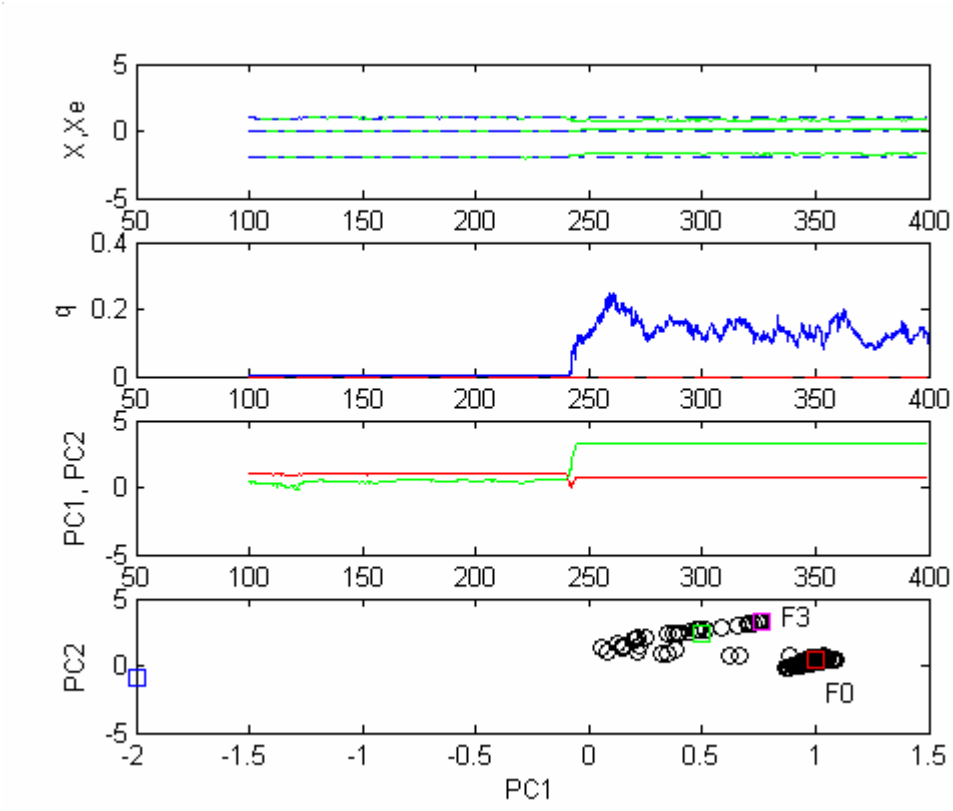


Fig. 5.41 - NLPCA inputs, scores, and SPE signal.

Remarks.

In this experiment, carried out with a nonlinear system, the neural approaches used for FDD showed a good performance, even in situations of a small time variant behaviour. The robustness property of the neural networks with respect to noise and non modelled dynamics is well known from relevant literature and from the results of many experiments, and here it has been confirmed once again.

In the FDD approaches based on system identification the performance of the controller influences the performance of the FDD approach. In the experiments, for the case of fault F_2, this is clear. For the faulty case F_2, an abrupt change of the design parameter r_0 (Eq. 5.10) of the adaptive LQG controller, from 0.4 to 0, causes oscillations on the system.

In this work, although the stability of switched systems was not investigated, it is certainly an important pointer for future research. Stability properties of a switched system in general depend on the properties of the switching signal. A switched system is stable if all individual subsystems are stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch (Liberzon, 2003). This stability subject is particularly important when dealing with adaptive systems, as in the case of this experiment where an adaptive optimal LQG controller is used. In this work, the LQG controller gains have been low pass filtered, in order to avoid abrupt transitions and also to try to prevent the loss of the plant controllability.

The interaction between controller and fault detection and diagnosis is also a subject that deserves special attention. For the FDD approaches based on the system identification this is particularly important, since the FDD methods require persistent excitation conditions and these conditions must be guarantee by the controller. For the case of adaptive controllers, also based on system identification, some faulty situations cause abrupt changes on the model parameters and this can provoke situations of failures. These failures are situations where the adaptive controller degrades its performance and can no longer guarantee a small control error. To solve some of these faulty cases is necessary to implement strategies based on fault tolerant control.

5.11 Fault Tolerant Control Experiments

In this section, some ideas related to Fault Tolerance and to Fault Tolerant Control (FTC) are presented first. This introduction to the fault tolerant control problem is very important for the experiment carried out with the three-tank benchmark to be understood. The ideas of fault detection and diagnosis are incorporated in this FTC context.

Later, a fault tolerant control approach is applied to the three-tank benchmark system, and some results are presented. The fault detection and diagnosis methodology based on a bank of neural recurrent output predictors (NROP), proposed in this dissertation, is used here in a context of fault tolerant control.

5.11.1 Introduction

First a definition of fault according to Blanke, et al., (2003), is given: a Fault in a dynamic system is a deviation of the system structure or the system parameters from the nominal situation.

In this section, the focus is on faults associated with structural changes. Structural faults change the set of the constraints and variables which are to be considered, as described in section 2.5.2. Examples of structural changes are the loss of a sensor, the blocking of an actuator (valve, etc), or the disconnection of a system component. In most situations, all these faults cause deviations of the dynamic input/output (I/O) properties of the plant from the nominal values, and change the performance of the closed-loop system.

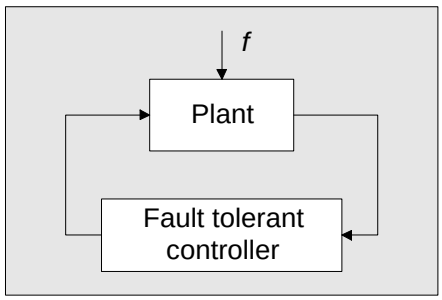


Fig. 5.42 - Fault tolerant system.

For a more detailed analysis of the impact of a fault let us consider the plant in Fig. 5.42 from the viewpoint of the controller. Here (in section 5.11), the term controller is used in a very

broad sense. The fault is denoted by f, and F is the set of all faults; the fault-free (faultless) case should also be included in the fault set F, and is denoted here by f_0. The pairs (**u**, **y**) are called input/output pairs (I/O pairs), i.e., the input and output process signals.

Blanke et al. (2003) define the behaviour B of a plant as the set of all possible pairs of trajectories **u** and **y** that may occur for the fault-free case. A graphical interpretation is given in Fig. 5.43. The behaviour B is a subset of the space U x Y of all possible combinations of input and output signals. The dot A represents an I/O pair consistent with the system dynamics, whereas the dot C is not consistent with the system dynamics.

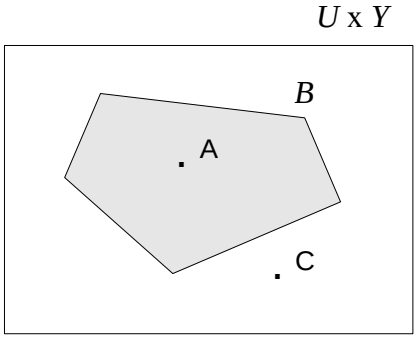


Fig. 5.43 - Graphical illustration of the system behaviour.

In discrete-time, the input **u** is represented by the sequence $\mathbf{u} = \{u(1),\ldots, u(k)\}$, and the output is described by the sequence $\mathbf{y} = \{y(1),\ldots, y(k)\}$. The behaviour is a subset of the Cartesian product, $B \subset R^k$ x R^k, which includes all sequences **u** and **y** that may occur for the fault-free plant. For dynamic systems, the I/O pair is a pair of (**u**, **y**) of sequences rather than a pair $(u(k), y(k))$ of current signal values.

Example 11. System behaviour of a static system.

Consider a static system described by the equation $y(k) = s_g\, u(k)$, where s_g is the static gain. The input and output are elements of the set R of real numbers. The set of all I/O pairs is given by $B = \{(u(k), y(k)) : y(k) = s_g\, u(k)\}$, which can be graphically represented as a straight line in the u/y coordinate system. Faults are found if the measured I/O pair $(u(k), y(k))$ does not belong to the behaviour B.

A fault changes the behaviour, as can be observed in Fig. 5.44 (Blanke, et al., 2003). In a fault-free situation the I/O pair $(u(k), y(k))$ belongs to the region of the behaviour B_0; when a fault occurs, the I/O pair $(u(k), y(k))$ moves towards the region of behaviour B_f. If a common

input U is applied to the fault-free and the faulty system, then both systems respond with different outputs Y_A and Y_B, respectively. The points A = (U, Y_A) and B = (U, Y_B) are located in different behaviour regions making the fault detection and isolation possible, unless the I/O pair (point C) lies in the intersection of B_0 and B_f.

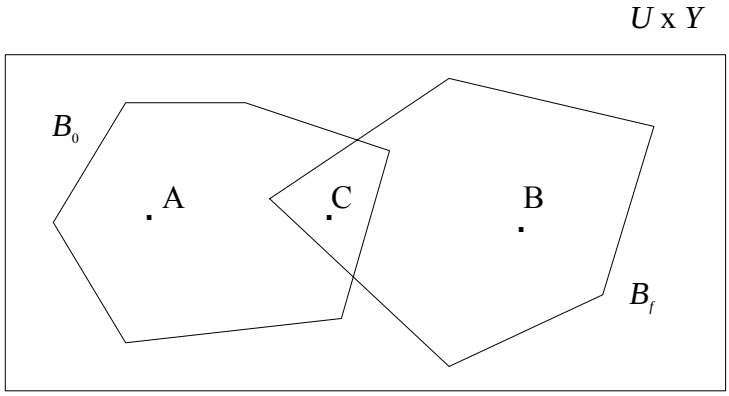


Fig. 5.44 - System subject to faults.

For Fault Tolerant Control (FTC), dynamic models of the plant subject to faults $f \in F$ have to be considered. These models describe the behaviour of the fault-free and the faulty system, i.e. they restrict the possible I/O pairs to those that appear in the behaviours B_0 and B_f in Fig. 5.44. So, models represent constraints that the signals $u(k)$ and $y(k)$ satisfy in order to be relevant for the plant (Blanke, et al., 2003).

For many decades engineers have investigated the faults impact, and different notions have been defined like safety, reliability, availability and dependability (Blanke, et al., 2003; Isermann & Balle, 1997). Safety describes the absence of danger; a safety system protects a technological system from permanent damage. Reliability is the probability that a system performs its intended function for a specified period of time under normal conditions; fault tolerant control (FTC) cannot change the reliability of the plant components, but it changes the reliability of the overall system. Availability is the probability of a system to be operational when needed; this system property depends on the maintenance policies. Dependability includes the three properties of safety, reliability and availability; a dependable system is a fail-safe system with high reliability and availability.

Due to its great importance, the relationship between fault tolerance and safety must be detailed. Assuming that the system performance can be described by two variables y_1 and y_2, in a two dimensional space, then different regions have to be considered as depicted in

Fig. 5.45 (Blanke, et al., 2003). During nominal operation, the controller must guarantee that the system remains in the region of required performance. In some situations, mainly when small faults occur, the controller "hides" the effects of faults, and this makes the detection and diagnosis tasks more difficult. A fault cause the system to move to the region of degraded performance, and the fault tolerant controller (FTC) should be able to initiate recovery actions. A safety system must interrupt the operation of the overall system so as to avoid danger, if the performance reaches the safety threshold. A fault tolerant controller and the safety system work in separate regions of the signal space, they are usually implemented in separate units, and that allows an independent design.

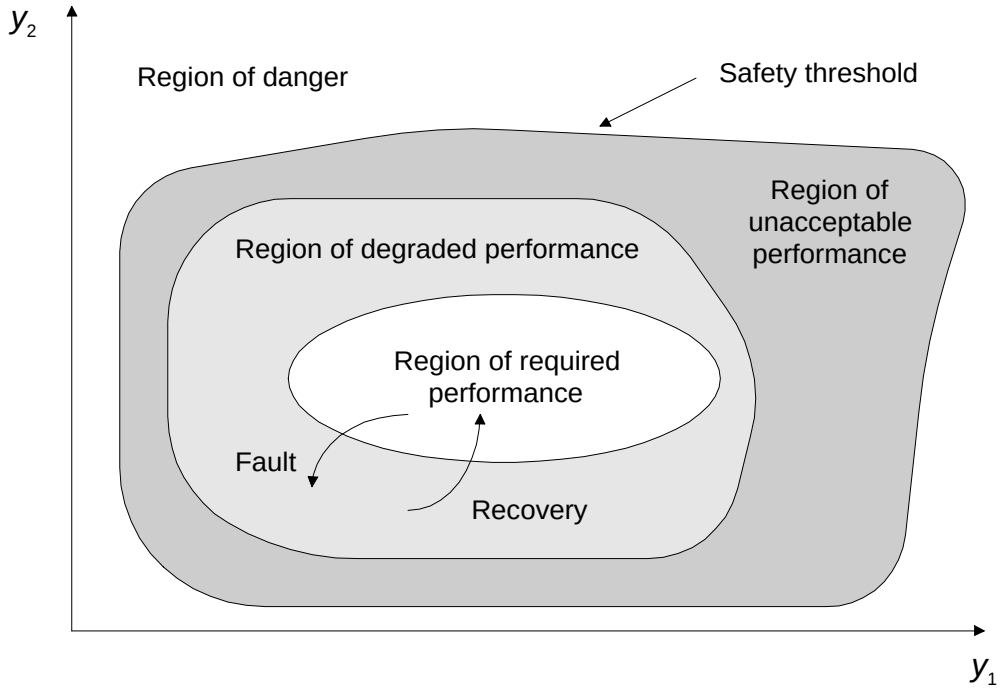


Fig. 5.45 - Performance regions.

5.11.2 Elements of Fault Tolerant Control

A typical architecture for Fault Tolerant Control (FTC) is depicted in Fig. 5.46 (Blanke, et al., 2003). At the supervision level, where there is a supervisor, the two blocks that carry out the two steps of fault tolerant control can be seen: the detection and diagnosis block, and the controller re-design block. The diagnostic block uses the measured I/O signals, or related features, and tests their consistency with the plant model. The re-design block uses the fault information, and adjusts the controller to the faulty situation.

Here (in section 5.11), the term controller is used in a very broad sense. The re-design of the controller may result in new controller parameters (re-tuning), or in new control architecture; the new and the old controllers may differ, not only with respect to the parameters, but also with respect to the input and output signals that they use.

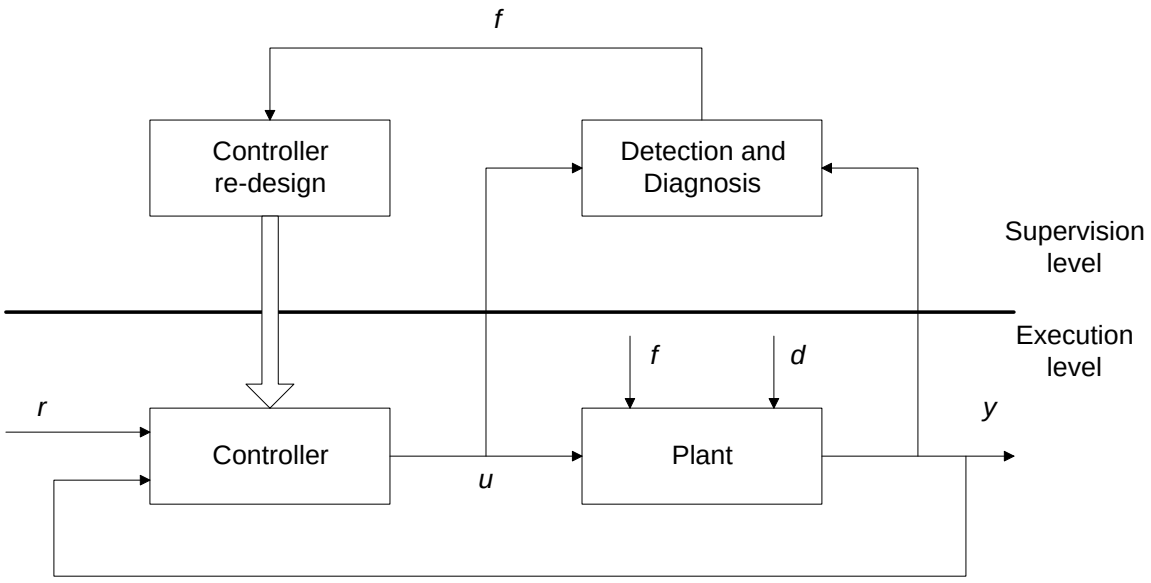


Fig. 5.46 - A typical architecture for fault tolerant control.

Two major types of methods for ensuring fault tolerance are well established: passive methods and active methods. To a certain extent, fault tolerance can also be attained without the structure given in Fig. 5.46, by means of control methods based on robust control and adaptive control; this is possible only for a restricted class of faults.

Passive fault tolerance can be obtained using a robust controller, where a fixed controller is designed to tolerate changes of the plant dynamics. This type of controller works suboptimally for the nominal plant because its parameters are obtained as a trade-off between performance and robustness (Patton, 1997).

Active fault tolerance can be obtained using adaptive control techniques (Astrom & Wittenmark, 1995; Mosca, 1995; Patton, 1997). In this case, the controller parameters are adapted to changes of the plant parameters, assuming the changes are caused by some fault. The theory of adaptive control shows that this principle is more effective if the plants are described by linear models and the faults cause slowly varying parameters.

If the faults cause severe effects that cannot be solved by robust and adaptive control techniques, then the ideas of fault tolerant control must be implemented.

Modern technological systems consist of several, often many subsystems, which are strongly connected. The effect of a fault in a single component is usually propagated throughout the overall system, as depicted in Fig. 5.47. If a fault causes the safety system to shut off the whole system, then the fault has caused a system failure.

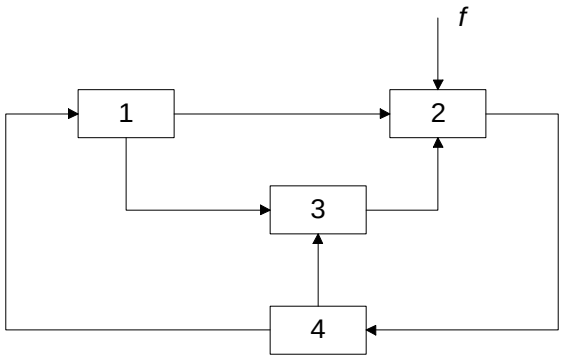


Fig. 5.47 - Fault propagation in interconnected systems.

The first task of Fault Tolerant Control concerns the Fault Detection and Diagnosis (FDD) of existing faults. Assuming that a dynamic system with input u and output y is subjected to some fault, f, the system behaviour depends on the fault $f \in F$, where the element f_0 of the set F symbolises the fault-free case. The FDD problem can be formulated as follows.

Problem 7. If the I/O pair $(\mathbf{u}, \mathbf{y})$ consists of sequences $\mathbf{u} = \{u(1),\ldots, u(k)\}$, and $\mathbf{y} = \{y(1),\ldots, y(k)\}$, then the fault detection and diagnosis (FDD) problem can be formulated as follows (Blanke, et al., 2003): for a given I/O pair $(\mathbf{u}, \mathbf{y})$, find the fault f.

For Fault Tolerant Control (FTC) the location and the magnitude of the fault have to be found. In summary, the fault detection and diagnosis principle can be described as follows (Blanke, et al., 2003). For given models that describe the behaviour B_f of the system subject to the faults $f \in F$, test whether the I/O pair $(\mathbf{u}, \mathbf{y})$ satisfies the relation $(\mathbf{u}, \mathbf{y}) \in B_f$.

Fault Detection: if the I/O pair is not consistent with the behaviour B_0 of the fault-free (faultless) system, $(\mathbf{u}, \mathbf{y}) \notin B_0$, then a fault has occurred.

Fault Diagnosis (isolation and identification): If the I/O pair is consistent with the behaviour B_f, $(\mathbf{u}, \mathbf{y}) \in B_f$, then the fault f may have occurred. Fault f is a fault candidate.

In a fault tolerant controller context, controller re-design considers the problem of changing the control structure and the control law after a fault has occurred in the plant. The goal is to

satisfy the closed-loop specifications. Two principal ways of controller re-design can be distinguished (Blanke, et al., 2003): fault accommodation, and control reconfiguration.
In fault accommodation, if there is a control law so that the faulty plant can achieve the control goal, then this law is used; this means to adapt the controller parameters to the dynamic properties of the faulty plant. If fault accommodation is impossible, then the control loop has to be reconfigured.
Control reconfiguration includes the selection of a new control architecture where alternative input and output signals are used. Control reconfiguration is necessary after the occurrence of severe faults that lead to serious structural changes of the plant dynamics, for example: a) sensor faults that break the information link between the plant and the controller, and make the plant partially unobservable; b) actuator faults that make the plant partially uncontrollable; c) plant faults that cannot be tolerated by any control law.

The main advantage of Fault Tolerant Control (FTC) over other fault tolerance methods is the fact that it makes intelligent use of the analytical redundancies included in the system, and the information about the system in order to increase the availability of the system (Blanke, et al., 2003). No method can guarantee a complete description of all possible system faults; hence no 100 % fault tolerance is possible. For many applications, only the most critical faults have to be investigated and tolerated. In some cases, FTC methods cannot be sufficiently tested in operation because in practice it is usually impossible to provoke some types of faults in the plant in order to test the reaction, so for these cases the simulation of faulty behaviours plays a very important role. Small faults are difficult to detect but easy to correct, whereas severe faults are easy to identify but difficult to correct.

5.11.3 FTC Approach applied to the Three-Tank Benchmark

The Three-Tank Benchmark has been elaborated within the COSY program of the European Science Foundation, in order to perform investigations of the control reconfiguration problem under severe structural faults.
The general problem to be solved is to find a new control strategy if a fault in the technical plant has occurred. More detailed information about the three-tank benchmark can be found in section 5.3.4. The severe faults (blocked valves and leaks) considered in this benchmark require the application of a fault tolerant control approach.

Many fault detection and diagnosis approaches, and fault tolerant control approaches, have been applied and tested on the three-tank benchmark, to deal with the reconfiguration problem. Some references using neural predictors, without a recurrent structure, are the papers of Koppen-Seliger, et al., (1999), and Marcu, et al., (1999).

A new fault tolerant control (FTC) approach is proposed here to solve the reconfiguration problem formulated for the three-tank benchmark. Let us remember here that a continuous-time model is used for the three-tank system, but all the algorithms have been implemented in this work in discrete-time. The three-tank benchmark is described in detail in section 5.3.4, and depicted here in Fig. 5.48 (Heiming & Lunze, 1999).

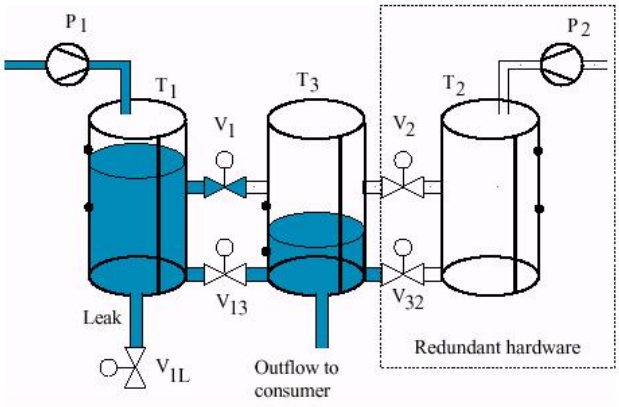


Fig. 5.48 - Three-tank system.

The fault tolerant control architecture used here follows the architecture depicted in Fig. 5.49 (Patton, 1997). This figure is shown for the case of a SISO system, without loss of generality. The thick lines represent signal flow, and the thin lines represent adaptation (tuning, scheduling or reconfiguration). The supervision system plays a crucial role in FTC applications. The supervisor must take decisions about adaptation when faults occur, in order to maintain the desired system performance and preserve the stability of the overall system. In most critical situations, the final decisions are taken by the humans.

In some non-severe faulty cases, the supervisor only needs to perform the re-tuning of the controller. When severe structural fault occurs, the supervisor usually needs to change the control strategy using other sensors, actuators, re-tuning the controllers and also changing the set-points.

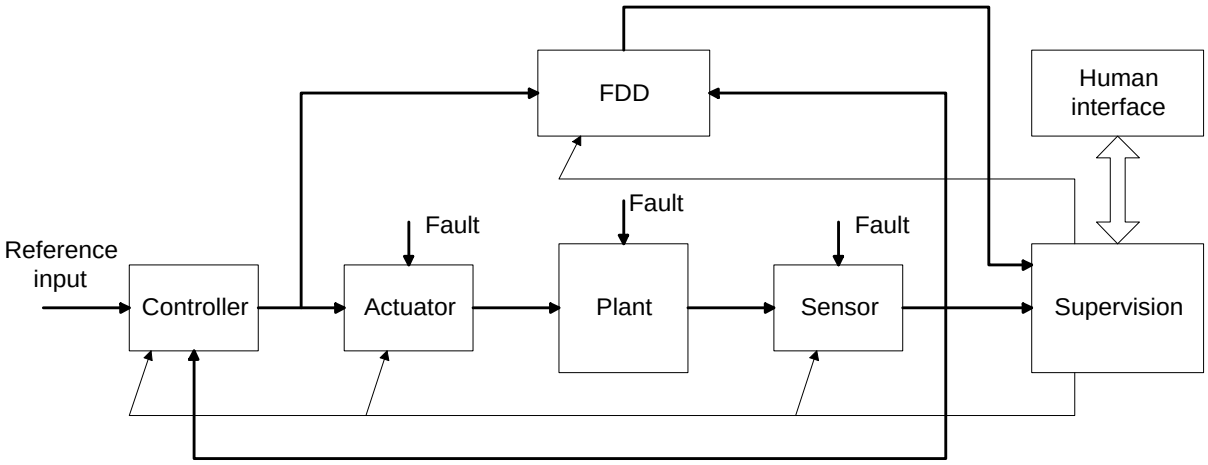


Fig. 5.49 - Fault tolerant control architecture.

The FTC approach used incorporates the FDD approach based on neural recurrent output predictors (NROP) proposed in this work (section 4.4). The FDD architecture is depicted again in Fig. 5.50.

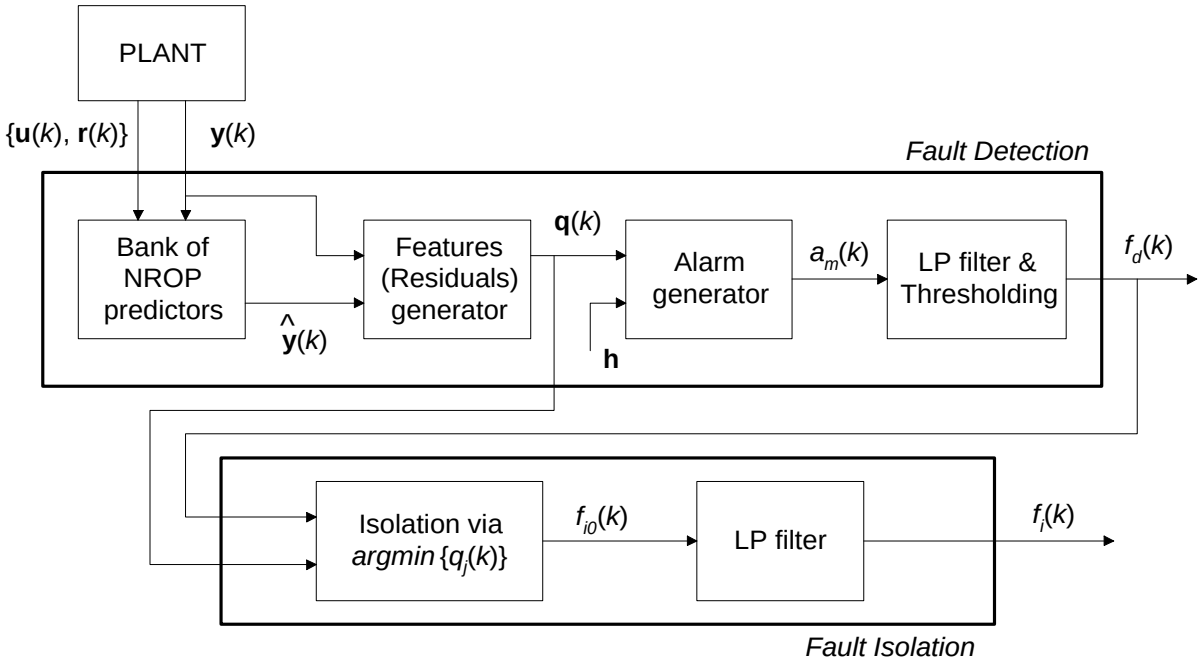


Fig. 5.50 - Architecture of the FDD approach based on a bank of NROP predictors.

In this problem a MISO system model is used, instead of the SISO model. The FDD approach is based on a bank of neural recurrent output predictors (NROP). One NROP is tuned to the nominal operating region, and the others are tuned to each faulty situation. A fault alarm $a_m(k)$ is generated if the square of prediction error (SPE), $q(k)$, associated in the NROP tuned to the nominal situation (fault F_0), exceeds the thresholds $\mathbf{h}$. The fault detection signal $f_d(k)$ is

obtained by low pass filtering $a_m(k)$ and by thresholding. The fault isolation signal $f_i(k)$ is computed based on the best tuned NROP predictor, i.e., the one with the smallest square of prediction error (SPE).

The reconfiguration problem for the three-tank benchmark is, in this work, formulated as follows.

Problem 8. For the three fault scenarios caused by severe faults, blocking valves and leak, find a new control strategy in order to maintain the main goal of the system. The main goal is to provide a continuous water flow Q_N to a consumer by maintaining the level $h_3(t) = medium$ in the central tank, T_3. The fault tolerant control system implemented must guarantee that severe faults do not cause failures, i.e., the interruption of the system's ability to perform the required goal.

The problem to be solved is the reconfiguration benchmark problem for the three-tank system, and the three fault scenarios are considered in the experiments (Heiming & Lunze, 1999). First, severe structural faults must be detected and diagnosed, and after it is necessary to find a new control strategy. The FTC systems usually need a supervisor system to take decisions.

The three-tank system is a multi-input multi-output (MIMO) system, and the main aim is to provide a continuous water flow Q_N to a consumer by maintaining the level $h_3(t) = medium$ in the central tank T_3. The level *medium* = [9; 11] cm.

The fault detection and diagnosis (FDD) methodology proposed here is based on a bank of neural recurrent output predictors (NROP) proposed in this work (section 4.4). For each fault scenario (F_0, F_1, F_2, F_3) a neural NROP predictor has been created. Each NROP predictor $\Psi_{Fi}(.)$ is tuned to the respective fault F_i, and expressed in the general form, for the case of SISO systems, by

$$\hat{y}_{nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}, \hat{y}_{nrop}(k\text{-}1), \ldots, \hat{y}_{nrop}(k\text{-}n), u(k\text{-}1), \ldots, u(k\text{-}n)) + K_n\, r_e(k\text{-}1)\,. \quad \text{Eq. 5.19}$$

The signal $y(k)$ is the output signal, $\hat{y}_{nrop}(k)$ is the output prediction signal, and $u(k)$ is the input signal.

The NROP predictor can also be applied to MIMO systems. The MIMO system is decomposed on a set of MISO systems, as described next. A MLP-FF neural network

implements a neural predictor model $NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W},\ldots)$. The predictor gain K_n has been chosen to guarantee stability and low predictor residuals.

The three-tank benchmark is a MIMO system (Fig. 5.48). For the fault scenarios of the reconfiguration problem, two MISO neural models are proposed here to implement the FDD approach based on NROP predictors. These neural models are based on output predictors of the system outputs $h_1(k)$ and $h_3(k)$, respectively, the water levels in tanks T_1 and T_3. Each neural predictor model embedded in the NROP predictor obeys the architecture depicted in Fig. 5.51, where the output $\hat{h}_i(k)$ represents either $\hat{h}_1(k)$ or $\hat{h}_3(k)$. The neural network inputs are the values for the sample k-1 of the water level in tank T_1, $h_1(k\text{-}1)$, the water level in tank T_3, $h_3(k\text{-}1)$, the flow from pump P_1, $q_1(k\text{-}1)$, and the discrete water level at tank T_3, $h_{3d}(k\text{-}1)$. All signals have been scaled to belong to the range [0; 1], except the signal $h_{3d}(k)$ that assumes discrete values {-1; 0; +1} corresponding to the water levels {*low*, *medium*, *high*} in the central tank. Each neural network predictor was trained off-line with data captured, in closed-loop, for each faulty situation, using the Levenberg-Marquardt optimization algorithm. Experiments with a duration of 600 s have been performed, and after each fault occurrence the data captured during 120 s was used for training the neural network embedded predictors. The number of neurons in each layer are, respectively, {IL, HL, OL} = {4, 4, 1}. After training during 400 epochs, using 3272 samples, SSE errors around 5×10^{-5} were reached for each neural network predictor.

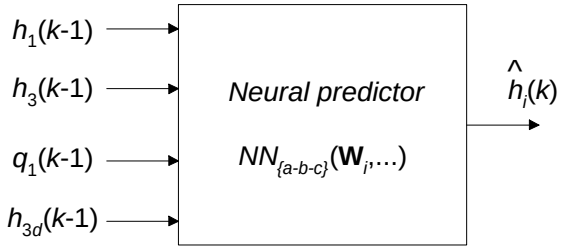


Fig. 5.51 - Architecture of neural output predictor embedded in the NROP.

Based on the neural output predictor, two neural recurrent output predictors (NROP) can be built, each one for the water levels in tanks T_1 and T_3. Assuming low order models, the equations for each neural recurrent output predictor (NROP) are given by

$$\hat{h}_{1\_nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}_1, \hat{h}_{1\_nrop}(k\text{-}1), h_3(k\text{-}1), q_1(k\text{-}1), h_{3d}(k\text{-}1)) + K_n\, r_e(k\text{-}1) \qquad \text{Eq. 5.20}$$

$$\hat{h}_{3\_nrop}(k) = NN_{\{a\text{-}b\text{-}c\}}(\mathbf{W}_3, h_1(k\text{-}1), \hat{h}_{3\_nrop}(k\text{-}1), q_1(k\text{-}1), h_{3d}(k\text{-}1)) + K_n\, r_e(k\text{-}1)\ . \qquad \text{Eq. 5.21}$$

In the experiments shown next, only the NROP predictor, $\hat{h}_{1\_nrop}(k)$, for the water level in tank T_1 has been used for FDD purposes. A bank of four NROP predictors of this type is used, each one tuned to a different faulty situation $\{F_0, F_1, F_2, F_3\}$. The gains for each NROP have been selected to obtain a good FDD performance, and they are given by {0.1; 0.1; 0.1; 1.2}.

In the experiments carried out with the three-tank benchmark model, at the start-up a PI controller was used to control the water level in tank T_1, and after the supervisor switch to an adaptive optimal LQG controller. The gains of the LQG controller are computed on-line based on a linearized ARX(2, 1, 1) model relating the water level $h_1(k)$) and the input flow $q_1(k)$. The sliding window SW-PCR algorithm is used for a window length of 20 s. A set point of 0.5 m is used to the level $h_1(k)$ in tank T_1, which corresponds to 0.5/0.6 = 0.83 in scaled values since 0.6 m is the maximum height of each tank. The gains selected for the PI controller are $K_p = 1$ and $T_i = 0.8$ s. The design parameter of the LQG controller is given by $r_0 = 0.03$, selected to guarantee stability for the different faulty cases and good persistent excitation conditions. For the central tank T_3, a switching (on-off) controller is used to guarantee a level $h_3(k)$ in the range [0.09; 0.11] m, i.e., [0.15; 0.18] in scaled values.
In the paper written by Heiming & Lunze (1999), where the definition of the three-tank benchmark problem and the typical fault scenarios are given, a PI controller is used to control the water level in tank T_1. In this work, an LQG controller is used. For the case of fault F_3, a PI controller is used to control the water level in tank T_2 (redundant hardware). The gains of the PI controller are the same used for the PI controller acting on the start-up on tank T_1, i.e., $K_p = 1$ and $T_i = 0.8$ s.

For the three-tank reconfiguration problem, the three typical fault scenarios and the proposed remedial actions are shown in Tab. 5.4. The nominal operation (without faults) corresponds to fault scenario F_0. Three typical faults scenarios are considered by Heiming & Lunze (1999): blocking valves and leaks, i.e., typical severe structural faults in the plant.
The first fault scenario F_1 is valve V_1 closed and blocked, and the aim is the water level in tank T_3 still *medium* (around 0.1 m). The fault F_2 is valve V_1 opened and blocked, and the aim is the water level in tank T_3 still *medium*. The last fault scenario, fault F_3, is a leak in tank T_1,

and the aim is to maintain the water level in tank T_3 still *medium* and guarantee a minimal loss of water from tank T_1.

Tab. 5.4 - Fault scenarios and proposed remedial actions for the three-tank benchmark.

Fault Scenario	Remedial Action
F_1: valve V_1 closed and blocked	Use valve V_{13} and pump P_1 to control water level in tank T_3. This is by achieved opening valve V_{13} and setting a new set-point of 0.2 m for the level $h_1(k)$, i.e., a scaled value of 0.2/0.6 = 0.33.
F_2: valve V_1 opened and blocked	Use pump P_1 to control water level in tank T_3. Here, this is performed by adjusting the set-point for the level $h_1(k)$ to the value 0.4 m (a scaled value of 0.4/0.6 = 0.67).
F_3: valve V_{1L} is open (simulating a leak in tank T_1)	Use of redundant hardware (tank T_2 and pump P_2). Here the following tasks are performed: a) use a PI controller to act on the pump P_2, with a set-point of 0.2 m for the level $h_2(k)$; b) open the valve V_{32}; c) maintain the valve V_{13} open while the condition $h_1(k) > h_3(k)$ is true.

All the three fault scenarios have been tested, and the results are presented later. Four experiments are shown. A fault is detected if the SPE computed based on the NROP residual exceeds the thresholds. The thresholds have been computed according to a three sigma limit approach given by $[\mu - 3\sigma, \mu + 3\sigma]$. Here, the mean value μ and the standard deviation σ obtained are given by $[\mu\ \sigma] = [1.2\times10^{-5}\ 2.4\times10^{-5}]$.

5.11.3.1 Experiment for fault F3 without reconfiguration

In the first experiment, the faulty scenario F_3 is tested, the fault is detected and diagnosed, and the system reconfiguration is not performed. The main idea, of this experiment without reconfiguration, is to understand what the main symptoms associated with this fault scenario are, and also to evaluate the performance of the FDD approach for the whole experiment. The other experiments are performed for each faulty scenario. After the correct fault detection and

isolation, the supervisor takes the appropriate actions to reconfigure the control system in order to tolerate the faults and guarantee that the severe faults do not cause failures, i.e., the interruption of the system's ability to perform the required goal.

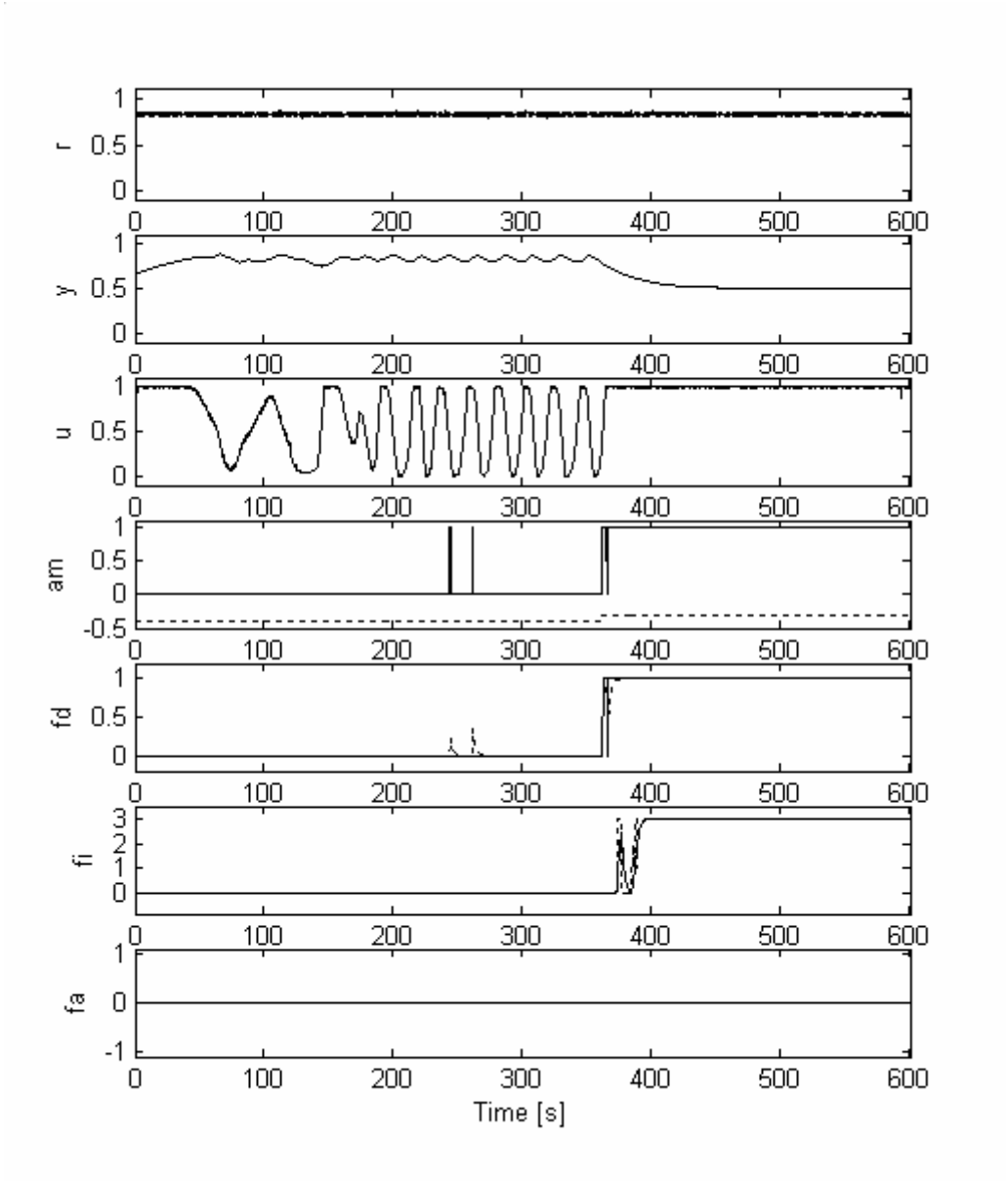


Fig. 5.52 - FDD signals for fault F3 without reconfiguration.

In Fig. 5.52, the input-output and FDD signals for the fault scenario F_3 can be observed, i.e., a leak in tank T_1 simulated by valve V_{1L} open. The reconfiguration is not performed here. This fault is the one that requires more reconfiguration actions, as described in Tab. 5.4.

A white noise with variance 1×10^{-8} has been added to the model output, simulating the sensor noise. The dynamics of the actuators (water pumps) and sensors have not been considered. The reference signal for level in tank T_1 is $r(k) = 0.83$, corresponds to a real value of 0.5 m, and a white noise (dither with variance 1×10^{-4}) was added to it. The signal $y(k) = h_1(k)$

represents the level on tank T_1, and $u(k)$ is the water flow from pump P_1, $q_1(k) = u(k)$, i.e., the input control signal. During the start-up and for up to 120 s, the PI controller is the active controller. Afterwards, the supervisor switches to the adaptive LQG controller. When the fault occurs the actuator saturates. In the figure, two false alarms can also be observed in the graph of $a_m(k)$, due to a violation of the thresholds for nominal operation. The fault occurs at time instant $t_k = 360$ s, as indicated by the trigger signal depicted in dotted line in the graph (label "am"). The last graphs show the fault detection signal $f_d(k)$, the fault isolation signal $f_i(k)$ and the fault analysis signal $f_a(k)$ (not considered here). The fault detection delay is 6.7 s, and the isolation delay is 36 s.

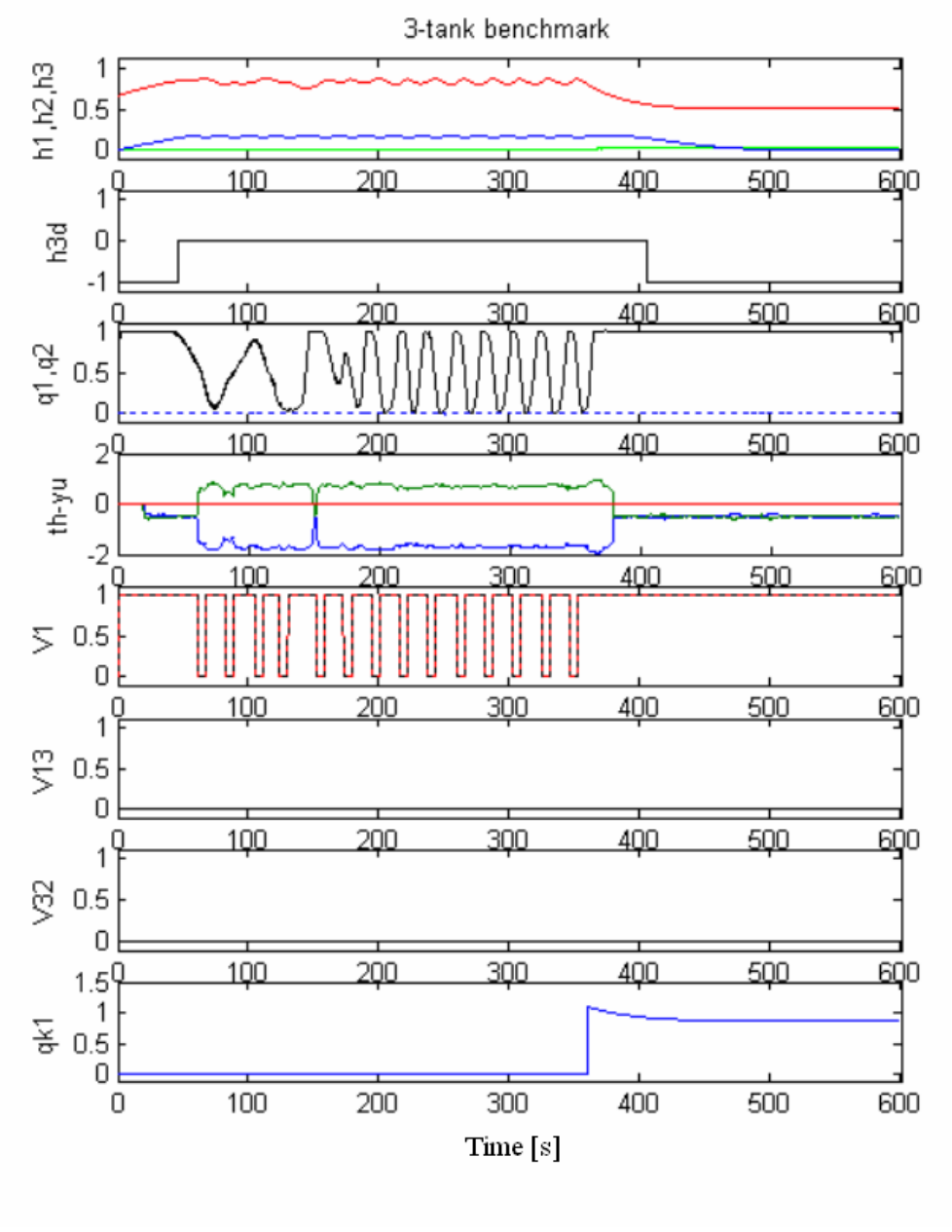


Fig. 5.53 - Flows and valves positions for fault F3 without reconfiguration.

Fig. 5.53 shows the main flows and valves positions. In the first graph the water level in each tank is depicted. Since, after the fault occurrence, the level $h_1(k)$ (red line) tends to a value of around 0.5, the level $h_3(k)$ (blue line) tends to a small value of around zero because the connecting pipes are placed at the bottom and at a height of 30 cm (at the middle of the tank). The next signal is the discrete level $h_{3d}(k)$ at the central tank T_3. The water flows from each pump are depicted in the next graph, $q_1(k)$ in solid line and $q_2(k)$ in dotted line. The ARX model parameters $\{a_1(k), a_2(k), b_1(k)\}$ (lines: blue, green and red), used by the adaptive LQG controller, are shown in the next graph ("th-yu"). The next three graphs show the positions of valves V_1, V_{13} and V_{32}. The last graph shows the water leakage flow $q_{k1}(k)$ through valve V_{1L}; this value has been scaled dividing it by the maximum flow Q_{max} for each pump.

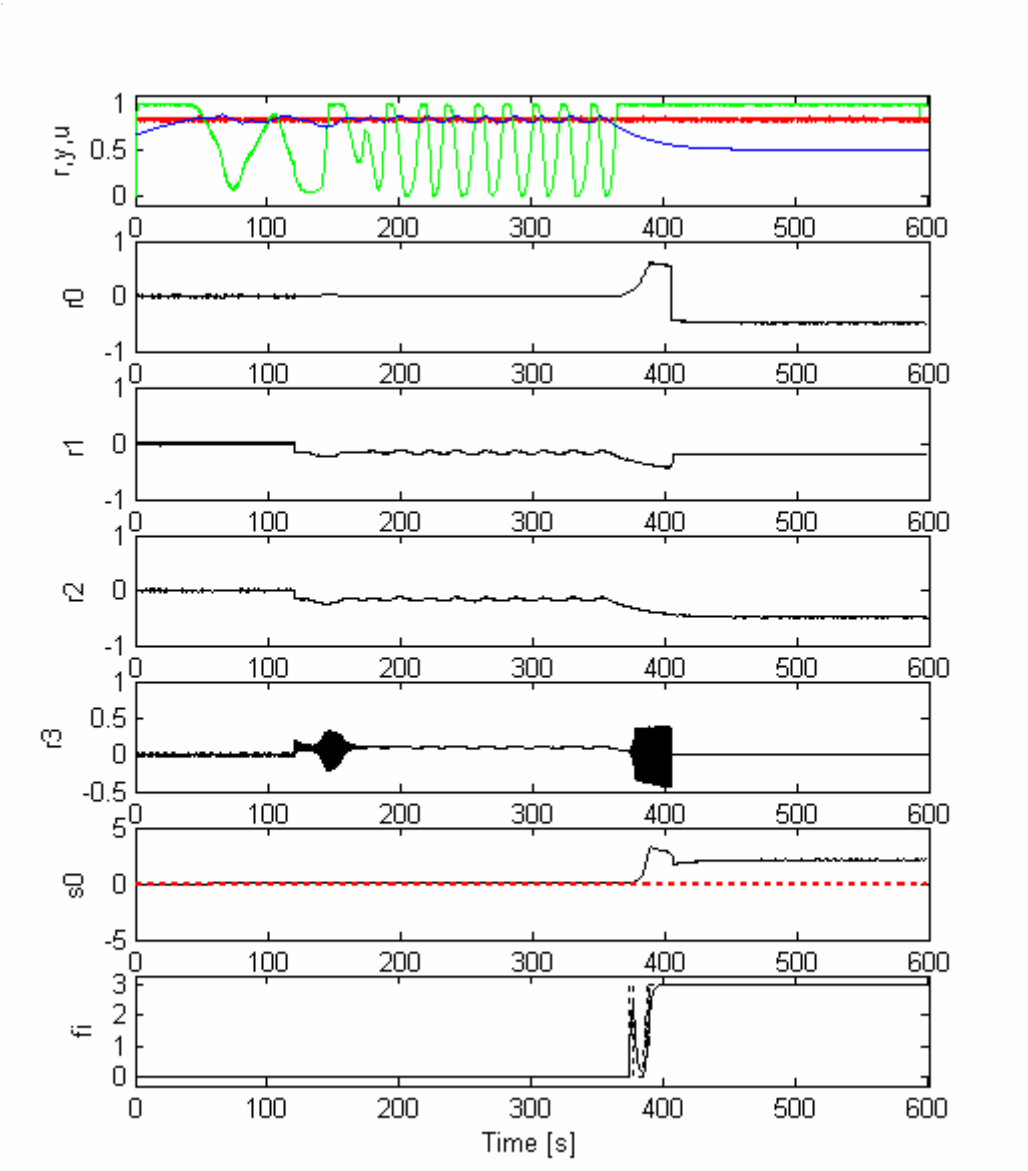


Fig. 5.54 - Residuals of NROP for fault F3 without reconfiguration.

Fig. 5.54 shows in the first graph the reference signal (red line) and the input-output signals (green line and blue line, respectively). Next the residuals $r_i(k)$ for each neural NROP predictor appear. In nominal operation, the residual $r_0(k)$ is approximately zero and the smallest. When fault F_3 occurs, the residual $r_3(k)$ becomes the smallest, and consequently its square of prediction error (SPE) also becomes the smallest.

A fault is detected if the SPE of the residual $r_0(k)$, i.e. the signal $q_0(k)$ (equal to $s_0(k)$) violates the thresholds (red lines). The fault isolation is performed according to the smallest SPE. The oscillations of the residual $r_3(k)$ of the NROP predictor tuned to fault F_3 occur because the NROP gain $K_n = 1.2$ is high, a value greater than the value ($K_n = 0.1$) of the others.

The next experiments show results for the reconfiguration problem for all the fault scenarios. In all cases, after fault detection and isolation the supervisor deactivates the FDD tasks until the end of the experiment. It is assumed that no more faults occur.

5.11.3.2 Experiment for fault F1 with reconfiguration

First, the results obtained for the fault scenario F_1, valve V_1 closed and blocked, are shown in Fig. 5.55.

From top to bottom, the reference signal $r(k)$, the output signal $y(k) = h_1(k)$, and the input signal $u(k) = q_1(k)$, all for tank T_1, in scaled values between [0; 1], can be observed. Next the fault alarm signal $a_m(k)$ and the fault detection signal $f_d(k)$ appear. The other graphs show the fault isolation signal $f_i(k)$, and the fault analysis signal $f_a(k)$ not evaluated here.

The fault occurs at time instant $t_k = 360$ s, the detection delay is 2.5 s, and the isolation delay is 20.1 s.

Some false alarms that provoke some false detections can be seen, but the isolated fault remains the fault F_0 (nominal behaviour) since the NROP associated has the smallest output residual.

Here, the true isolation only is performed if the isolation signal, $f_i(k)$, is inside a band of 5% of the isolated fault value during 5 s; this strategy avoids wrong isolations that occur at the beginning of some faults. This means that the true fault isolation occurs at time instant $t_k = 380.1$ s.

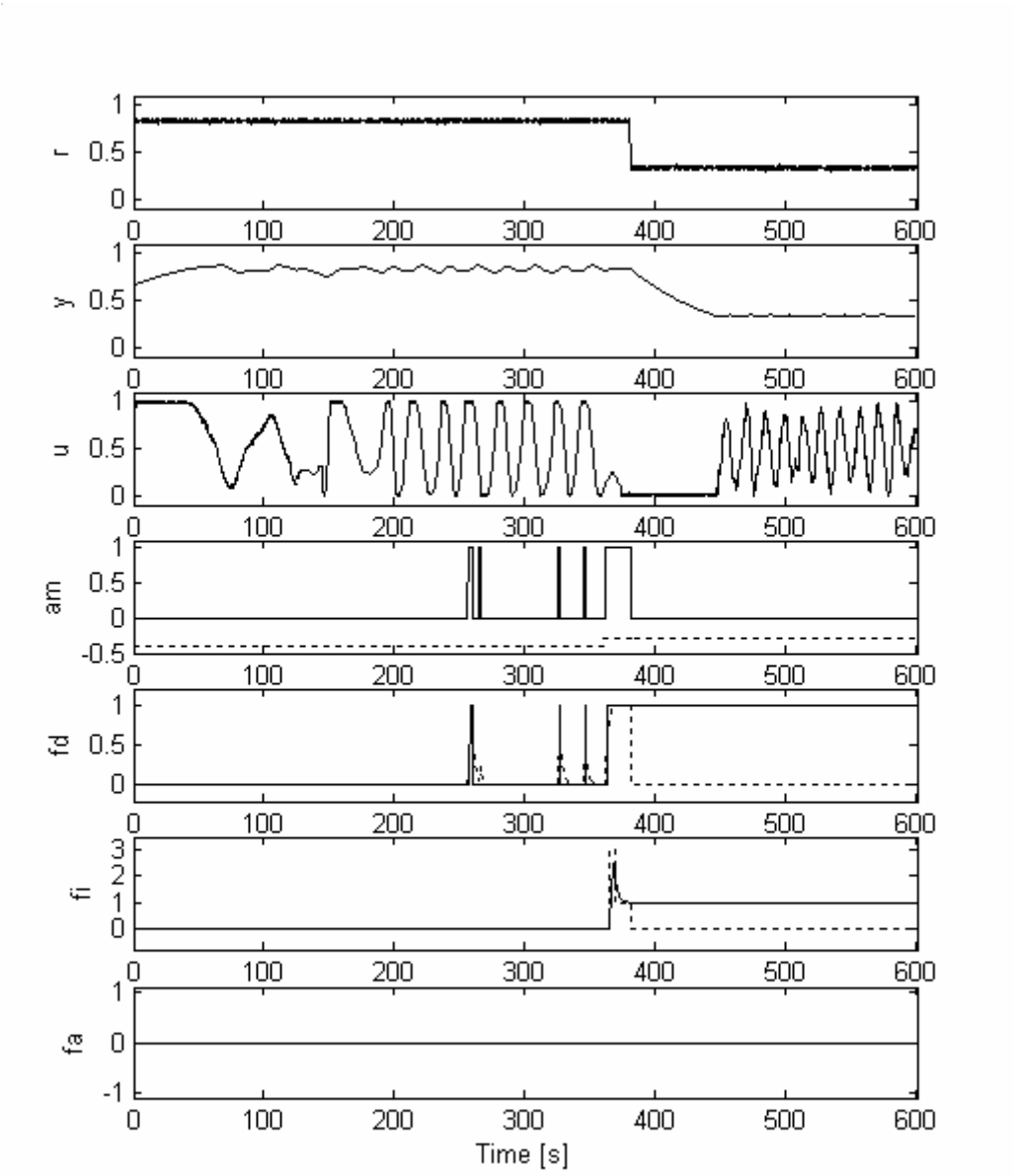


Fig. 5.55 - FDD signals for fault scenario F1 with reconfiguration.

The system reconfiguration takes place immediately after fault isolation, i.e., around 380.1 s. The reconfiguration tasks are the setting of a new set-point (0.2 m) given by 0.33 in scaled value, and to open the valve V_{13}, as depicted in Fig. 5.56.

In the first graph the water level in each tank is depicted. Each water level $h_1(k)$, $h_2(k)$ and $h_3(k)$ is represented, respectively, by the red line, the green line and the blue line. The next signal is the discrete level $h_{3d}(k)$ at tank T_3. The water flows from each pump are depicted in the next graph, $q_1(k)$ in solid line and $q_2(k)$ in dotted line. The ARX model parameters $\{a_1(k), a_2(k), b_1(k)\}$, used by the adaptive LQG controller, are shown in the graph with label th-yu. The next three graphs show the positions of valves V_1, V_{13} and V_{32}. The last graph

shows the water leakage flow $q_{k1}(k)$ through valve V_{1L}; this value has been scaled dividing it by the maximum flow Q_{max} of each pump.

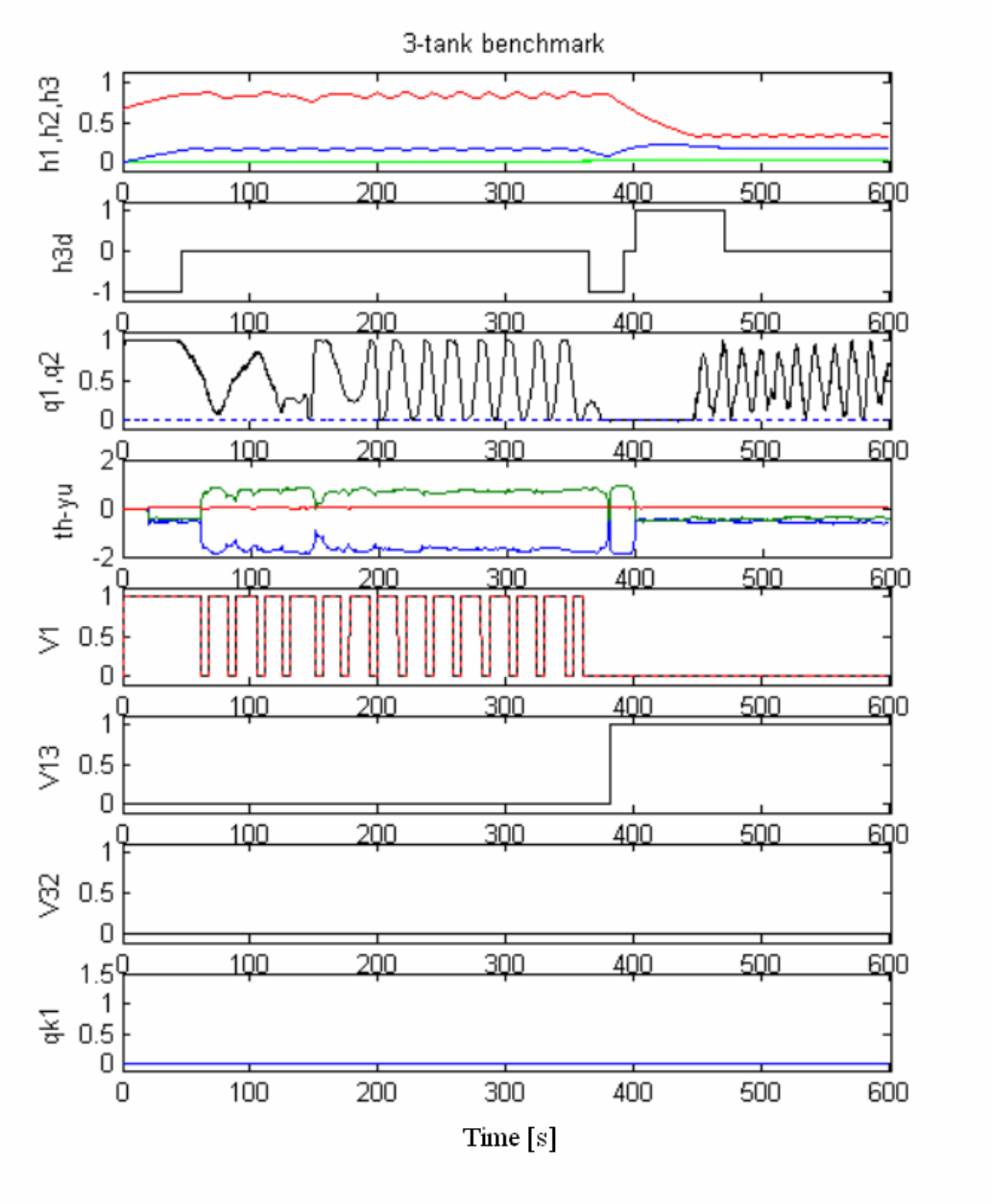


Fig. 5.56 - Flows and valves positions for fault F1 with reconfiguration.

For the fault scenario F_1 the results obtained are good, since the fault tolerant control system tolerates the severe fault and guarantees that this fault does not cause a failure, i.e., the interruption of the system's ability to perform the required goal. In fact, the water level in the central tank T_3 never reaches the zero value and this guarantees the water supply for the consumers.

5.11.3.3 Experiment for fault F2 with reconfiguration

The results obtained for the fault scenario F_2 are presented later.

For the fault scenario F_2, valve V_1 open and blocked, the input-output and FDD signals are depicted in Fig. 5.57. From top to bottom can be seen the reference signal $r(k)$, the output signal $y(k) = h_1(k)$, and the input signal $u(k) = q_1(k)$, all for tank T_1, in scaled values between [0; 1]. Next the fault alarm signal $a_m(k)$ and the fault detection signal $f_d(k)$ appear. The other graphs show the fault isolation signal $f_i(k)$, and the fault analysis signal $f_a(k)$ not evaluated here.

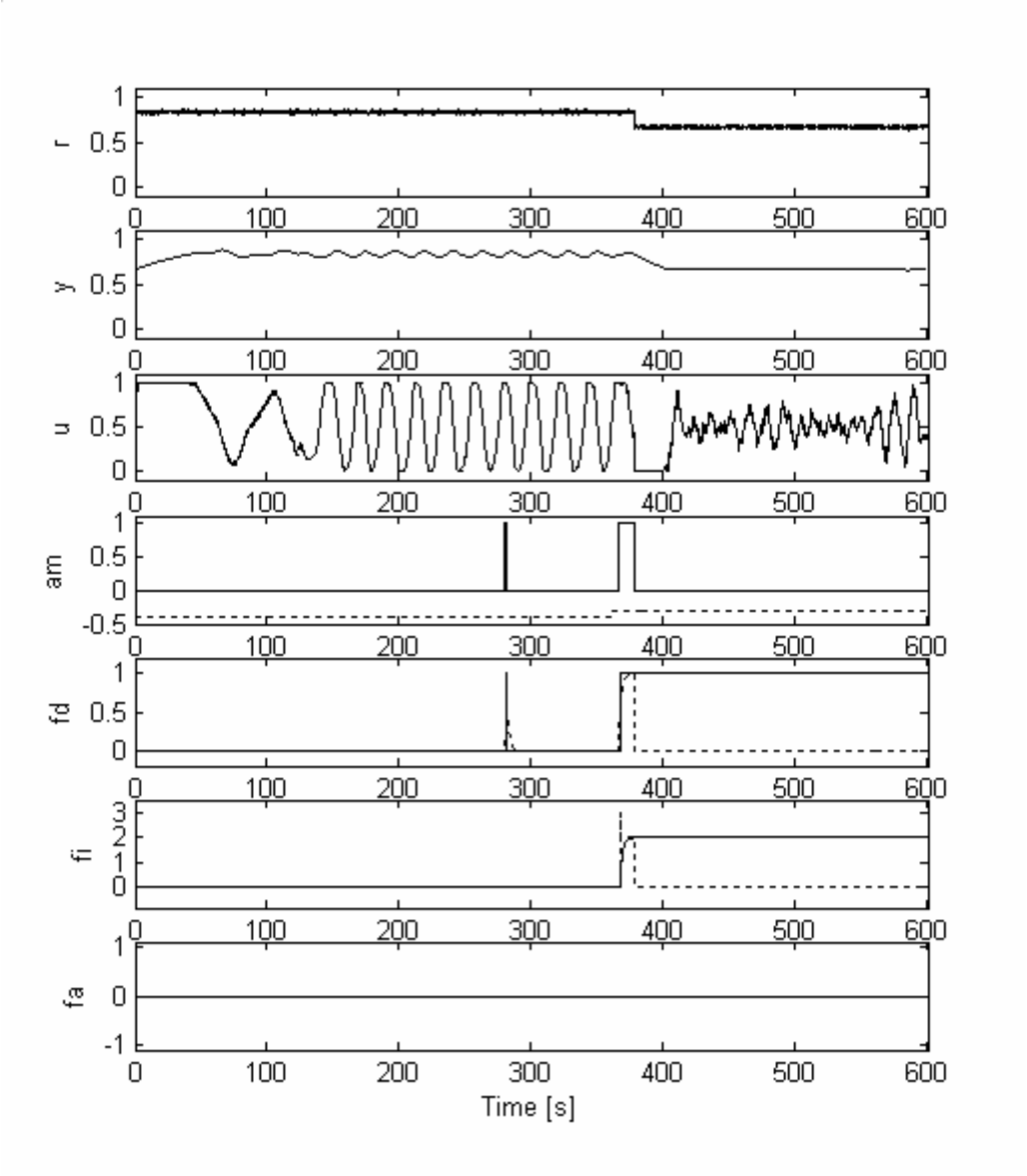


Fig. 5.57 - FDD signals for fault scenario F2 with reconfiguration.

The fault occurs at time instant $t_k = 360$ s, the detection delay is 7.1 s, and the isolation delay is 18.1 s. Here, the true isolation only is performed if the isolation signal, $f_i(k)$, is inside a band

of 5% of the isolated fault value during 5 s; this strategy avoids wrong isolations that occur at the beginning of some faults. This means that the true fault isolation occurs at time instant t_k = 381.1 s. After fault detection and isolation the supervisor deactivates these tasks until the end of the experiment.

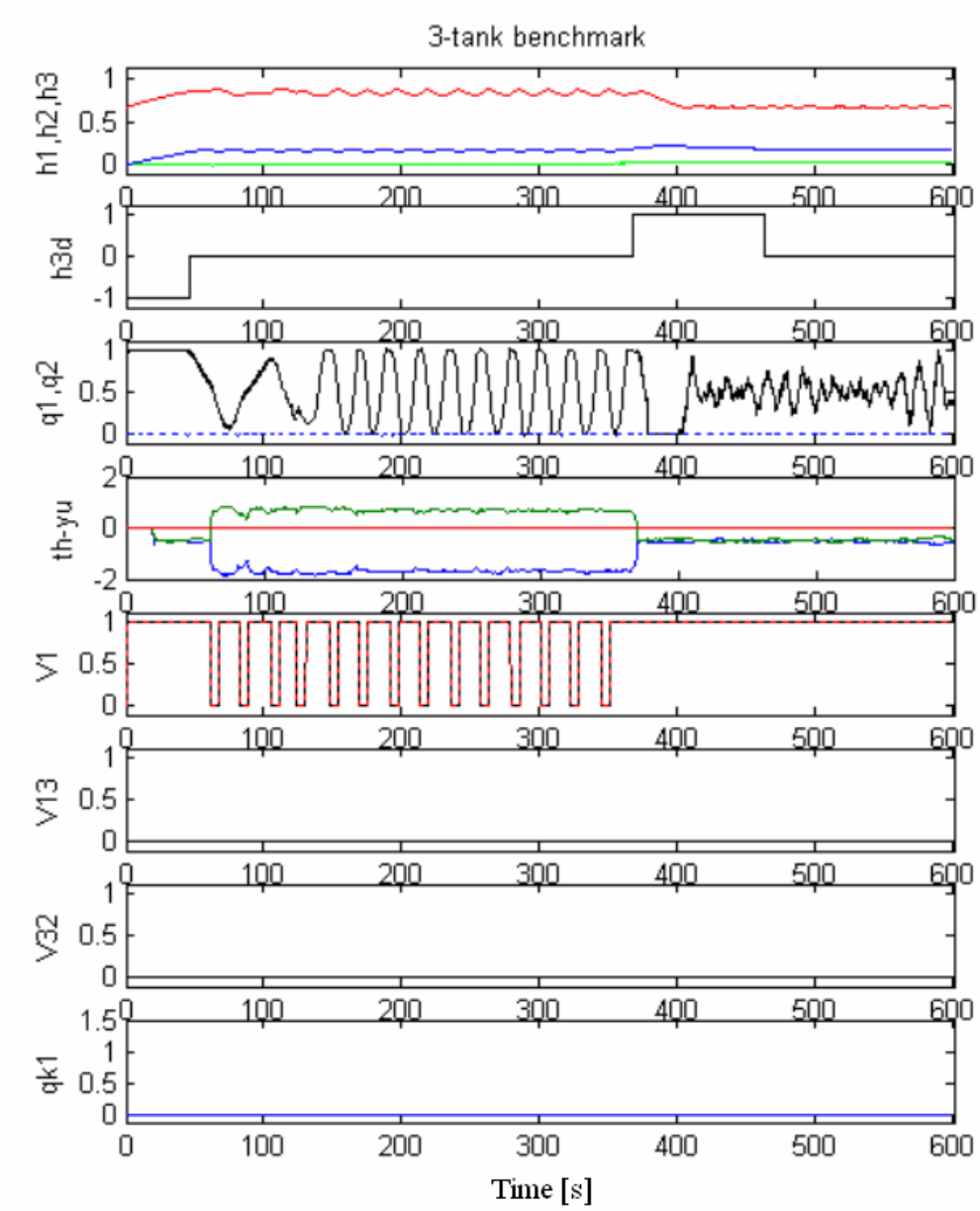


Fig. 5.58 - Flows and valves positions for fault F2 with reconfiguration.

The system reconfiguration takes place immediately after fault isolation, i.e., around 381.1 s. The reconfiguration tasks are the setting of a new set-point (0.4 m) given by 0.67 in scaled value, as depicted in Fig. 5.58. In the first graph the water level in each tank is depicted. Each water level $h_1(k)$, $h_2(k)$ and $h_3(k)$ is represented, respectively, by the red line, the green line and the blue line. The next signal is the discrete level $h_{3d}(k)$ at tank T_3. The water flows from

each pump are depicted in the next graph, $q_1(k)$ in solid line and $q_2(k)$ in dotted line. The ARX model parameters $\{a_1(k), a_2(k), b_1(k)\}$, used by the adaptive LQG controller, are shown in the graph with label "th-yu". The next three graphs show the positions of valves V_1, V_{13} and V_{32}. The last graph shows the water leakage flow $q_{k1}(k)$ through valve V_{1L}; this value has been scaled dividing it by the maximum flow Q_{max} of each pump.

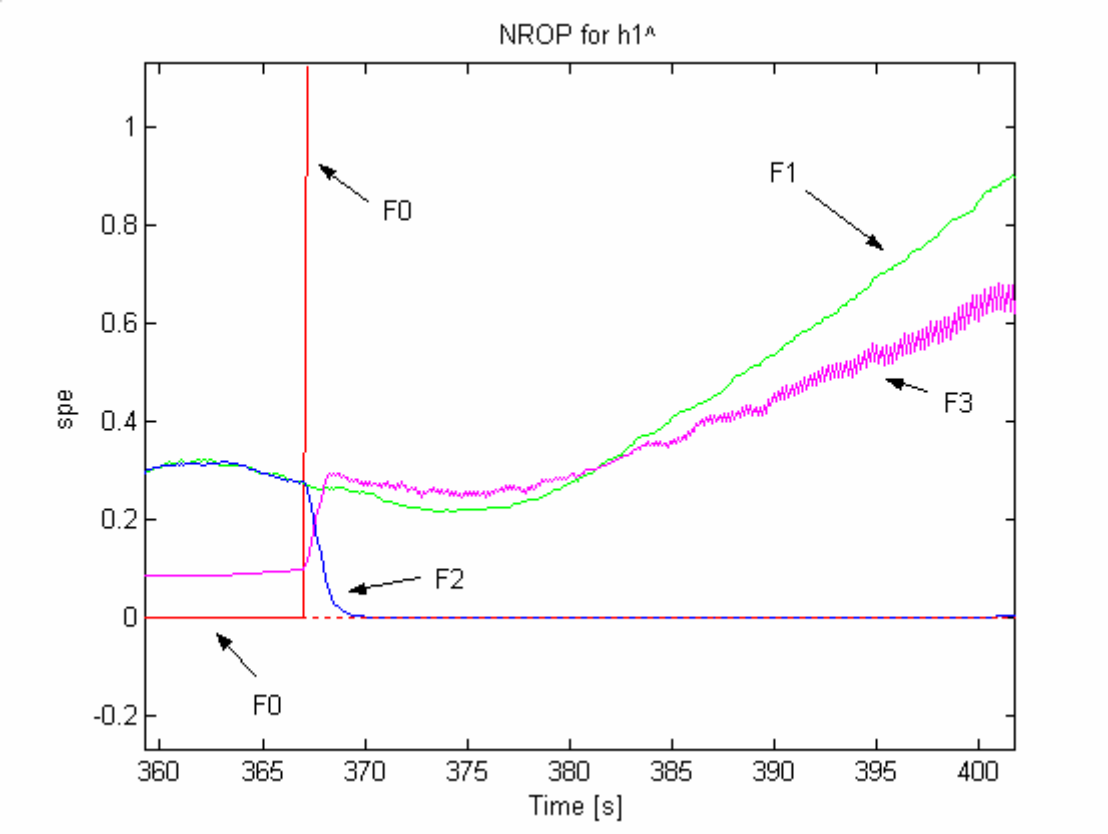


Fig. 5.59 - SPE signals for fault F2 with reconfiguration.

The FDD approach used in this experiment is based on a bank of neural recurrent output predictors (NROP), and the detection occurs when the SPE signal, computed from the NROP residual, exceeds the thresholds. The isolation criterion is given by the smallest SPE signal. The SPE signals for all NROP residuals are shown in Fig. 5.59. It can be seen that, a certain time after fault occurrence, the SPE signal (red line) associated with the NROP predictor tuned to fault F_0 changes from a small value, around zero, to a value greater than one, and consequently the fault is detected since the thresholds are exceeded. The SPE signal (blue line) for the NROP tuned to fault F_2 tends to zero, becomes the smallest and consequently fault F_2 is the isolated fault. As expected, the SPE signal (magenta line) for fault F_3 exhibits the highest variance since the associated NROP gain is given by $K_n = 1.2$, a value much greater than the other gains ($K_n = 0.1$).

5.11.3.4 Experiment for fault F3 with reconfiguration

The results obtained for the fault scenario F_3 are shown later. For the fault scenario F_3, valve V_{1L} is open (simulating a leak in tank T_1). In Fig. 5.60 the input-output and the fault detection and isolation signals are depicted. For this fault it is necessary to use the redundant hardware (tank T_2 and pump P_2). The first graph shows the reference signal $r(k)$. Next the output (level) signal $y(k) = h_1(k)$ and the input (control) signal $u(k) = q_1(k)$ for tank T_1 appear. Then the fault alarm signal $a_m(k)$, the fault detection signal $f_d(k)$, and the fault isolation signal $f_i(k)$ are shown. The last signal $f_a(k)$ is zero because the fault analysis is not performed here. The fault occurs at time instant $t_k = 360$ s, and the detection delay is 6.5 s. The isolation delay is 42.5 s, i.e., fault is well isolated at time $t_k = 402.5$ s.

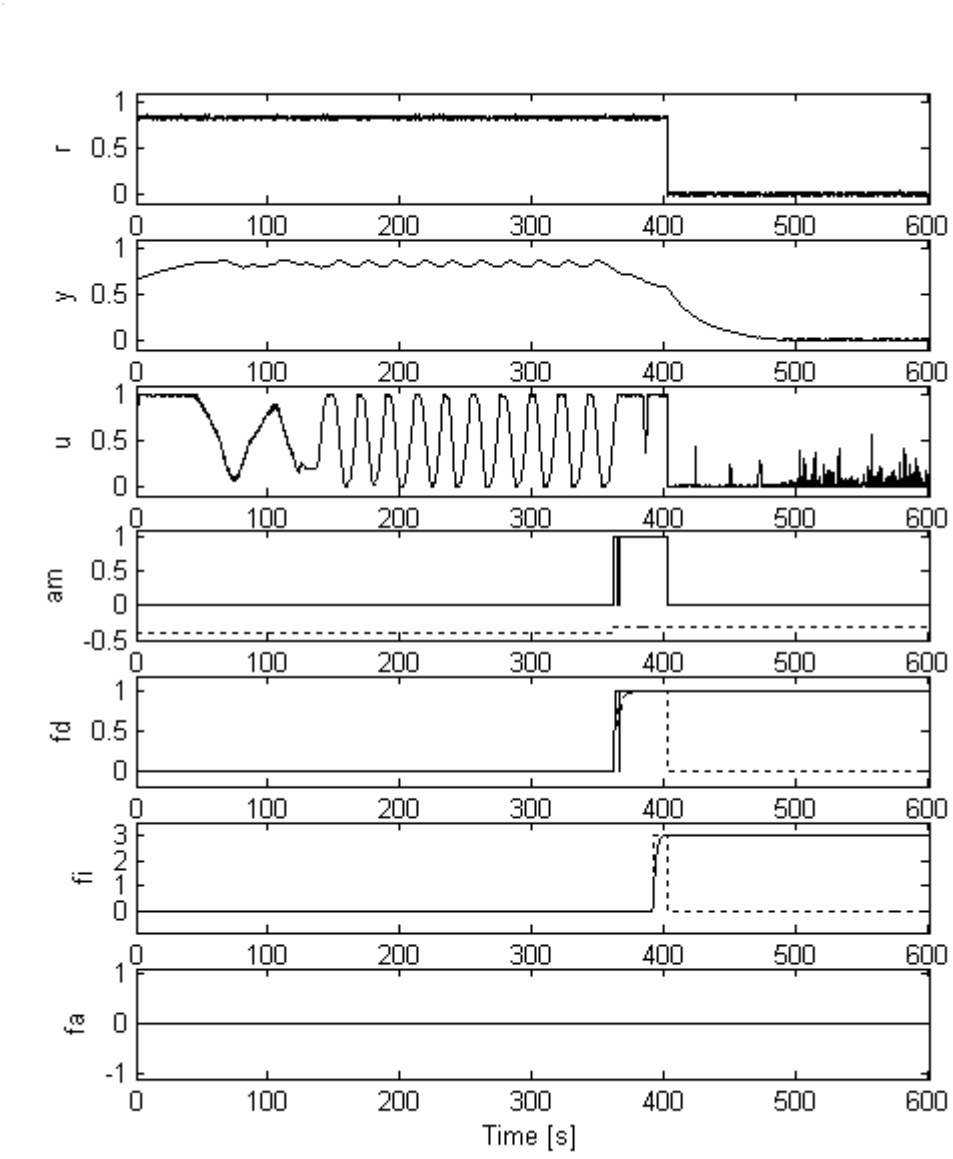


Fig. 5.60 - FDD signals for fault scenario F3 with reconfiguration.

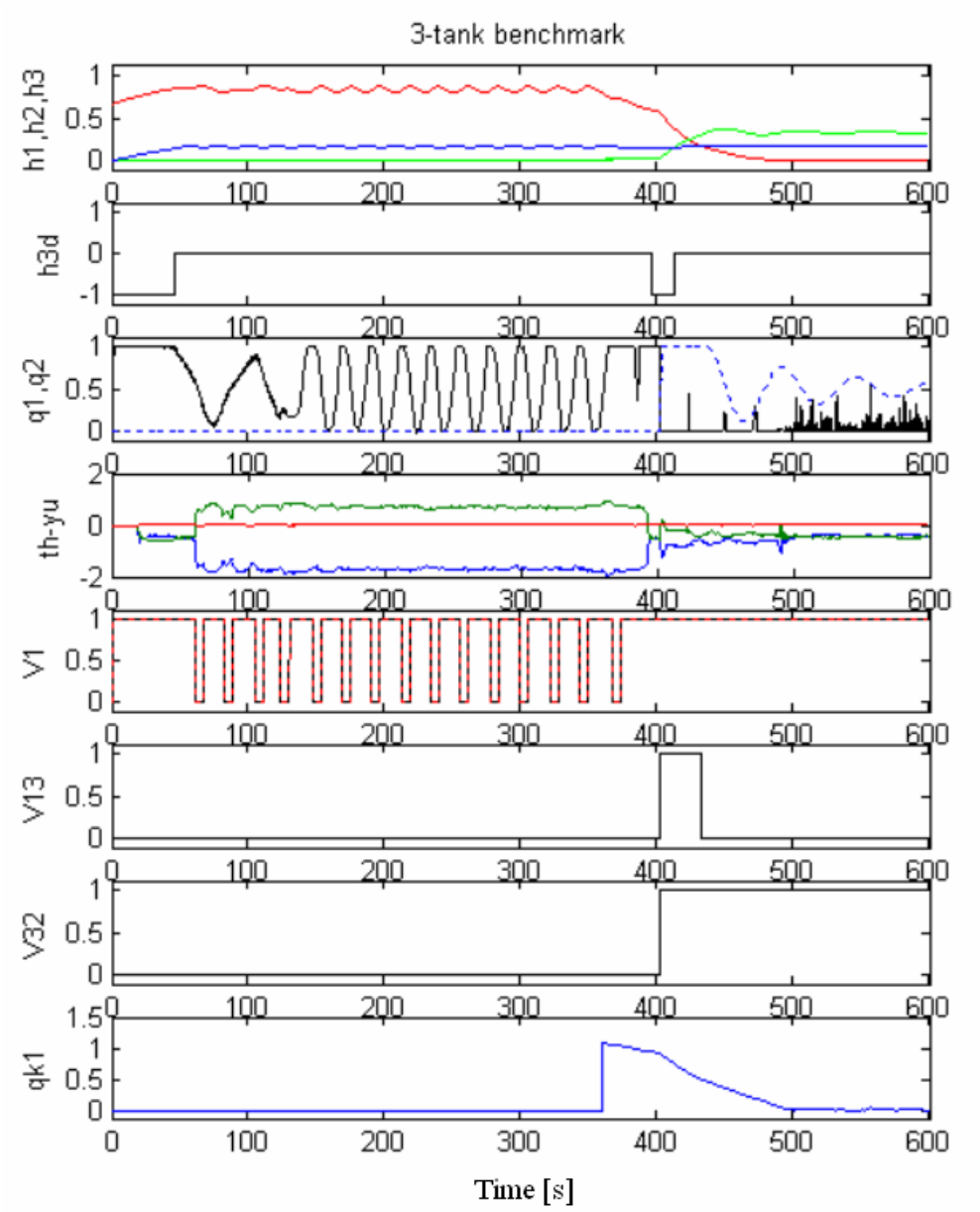


Fig. 5.61 - Flows and valves positions for fault F3 with reconfiguration.

In Fig. 5.61 the flows and the valves positions signals are depicted. Each water level $h_1(k)$, $h_2(k)$ and $h_3(k)$ is represented, respectively, by the red line, the green line and the blue line. Since there is a leak in tank T_1, the supervisor adjusts the level set-point on this tank to zero after fault isolation. Since a small white noise dither is added to the set-point signal, this means that the new set-point for tank T_1 is not absolute zero, but a small value near zero. This dither is enough to guarantee the persistent excitation conditions for the LQGC, as seen in the graph "th-yu" of the ARX model parameters. This explains the small control action, $q_1(k)$ (solid line), that appears on the third graph. The supervisor activates the redundant hardware, and consequently a PI controller becomes active to control the level $h_2(k)$ of tank T_2. The third

graph also shows the control action, $q_2(k)$, in dotted (blue) line. The gains of the PI controller are the same used for the PI controller acting on the start-up on tank T_1, i.e., $K_p = 1$ and $T_i = 0.8$ s. The other graphs show the valves positions and, finally, the last graph shows the leak flow through the valve V_{1L} that tends to zero.

Remarks.

The fault tolerance in dynamic systems is a very important subject in critical systems like nuclear plants, communications systems, aeronautics, chemical and power plants, supplying systems, etc. In fact, the three-tank benchmark system emulates a critical system, i.e., a supplying system.

The control reconfiguration problem under severe structural faults, considered in this section, belongs to the set of complex problems in the FTC research area, since severe faults are easy to identify but difficult to correct.

The experiments carried out with the three-tank benchmark show a good FDD/FTC performance. For each fault, the detection delay and the isolation delay are reasonable due to the slow dynamics of the three-tank system. The values are summarized in Tab. 5.5.

Tab. 5.5 - Detection delays and isolation delays for all faults.

	Detection delay [s]	Isolation delay [s]
Fault scenario F_1	2.5	20.1
Fault scenario F_2	7.1	18.1
Fault scenario F_3	6.5	42.5

The fault detection and diagnosis approach based on a bank of neural recurrent output predictors (NROP), proposed in this work, reveals a good performance to deal with these severe structural faults, and shows to be efficient to deal with MISO systems.

The fault tolerant control strategy proposed in this work for the three-tank benchmark problem has shown to be effective. The adaptive optimal LQG controller performs well for the different fault scenarios, using a fixed design parameter, $r_0 = 0.03$. A better control performance is expected if the design parameter, r_0, is adjusted on-line to each faulty scenario. The dither on the reference signal has been used to guarantee persistent excitation conditions, needed for the on-line estimation of the ARX model used by the LQG controller.

5.12 Conclusions

The experiments are a great help in understanding and validating the theory. The practice helps to understand the advantages and drawbacks of the theory better. In research, both theory and practice must interact for mutual development.

In this chapter, experimental results applying the fault detection and diagnosis approaches proposed in this dissertation have been presented. The experiments have been performed using process models and a real nonlinear DC motor setup. In this work, many examples were given based on simulation models. Mainly, three reasons support this option: a) the simulation models allow the validation of the approaches and algorithms; b) the simulation models offer more flexibility for faults testing; c) the simulation saves considerable time.

The various experiments performed throughout the work have shown that black-box models have a great potential for fault detection and diagnosis. Most industrial plants are complex systems, and consequently white box models are not available, or are hard to obtain. In the majority of plants, the FDD approaches based on black-box models are the only solution for FDD purposes. In this work the new FDD approaches proposed are based on linear ARX models and nonlinear NARX neural models.

The FDD approaches have been explained assuming the existence of a single fault, but they can also be applied to the case of simultaneous faults.

The assumption of normality for the data is particular important to compute thresholds used for fault detection and fault diagnosis. In this work, the assumption of normality for the FDD features signals, under nominal operating conditions, is a reasonable assumption. In this work, the fault detection and isolation signals are low pass filtered to reduce the false alarm rates. If necessary, the thresholds obtained assuming the data is normally distributed can be experimentally adjusted to guarantee a reasonable FDD performance in terms of ratio of false alarms.

The performance of the FDD approaches proposed depends strongly on the quality of the black-box models. So, efficient system identification methods are necessary. The proposed sliding window SW-PCR parameter estimation algorithm proved to be efficient, and better for FDD than the recursive RLS algorithm. The on-line identification of ARX process models requires good persistent excitation conditions (PEC). To guarantee the PEC a dither signal has been added to the reference signal, provoking an increase on the control error; typical values obtained for the maximum control error are around 4 % of the set-point signal, for a set-point in the range [0; 1].

The approaches based on on-line system identification use the sliding window SW-PCR parameter estimation algorithm. Since for a sliding data-window with length τ, one knows that the transient following a parameter jump lasts exactly τ - 1 samples, then the isolation delay cannot be smaller than the window length of the algorithm SW-PCR. In the experiments a window length of 10 s was used, except for the case of the three-tank benchmark where the value is 20 s.

The FDD approaches based on nonlinear NARX neural models presents a good performance. Nonlinear FDD is usually more difficult than linear FDD, since nonlinear models are hard to obtain. The neural networks have shown a great potential to perform the different tasks of output prediction, nonlinear principal components analysis and nonlinear discriminant analysis. In fact, they are a key element in this work, since they are used in most of the new FDD approaches proposed.

In the literature, some authors argue that typically the fault detection task is simpler than the fault diagnosis task. In this work, the experiments confirm this statement.

The results obtained with the proposed fault detection and diagnosis approach, for SISO systems, based on dynamic features (static gain and bandwidth) of ARX models have shown a good performance, for the class of faults tested on a model of a DC motor. The greater number of faults tested have been multiplicative (parametric) faults, but this approach can also detect and diagnose additive faults. Both input-output ARX models $M_{yu}(\theta)$ and reference-output ARX models $M_{yr}(\theta)$ can be used for FDD. There are two main advantages of models $M_{yr}(\theta)$: a) enables faults on the process and also on the controller to be detected; b) usually exhibits smaller variances on the estimated parameters.

The sliding window PCR algorithm presents a good performance, since the estimates are consistent. The performance of this type of approach based on system identification depends strongly on the the variance of the estimated parameters of the ARX model. This variance is a function of the sensor noise variance and also of the persistent excitation conditions. The variance of the parameters increases if there is an increase on the sensor noise variance and/or a degradation of the persistent excitation conditions.

In the experiments performed the bandwidth signal shows a greater variance than the static gain signal. The estimated static gain has shown to be a better feature for fault isolation.

In this approach, the features space for FDD is a two dimensional features space given by the static gain and the bandwidth. The neural nonlinear discriminant analysis (NNLDA) gives good results for pattern classification used for fault detection and isolation.

The fault detection and diagnosis approach based on the combined approach PCA & IMX has been tested on a DC motor model. Both input-output ARX models $M_{yu}(\theta)$ and reference-output ARX models $M_{yr}(\theta)$ can be used for FDD.

PCA is a good technique for fault detection, but for diagnosis statistical techniques (Fisher Discriminant Analysis, NNLDA) are more appropriate since they take into account the information between the classes. The results obtained with the fault detection approach based on PCA applied to ARX model parameters are reasonably good, taking into consideration that the data does not obey a perfectly normal distribution. The deviations from normality require an adjustment of the threshold computed for the *T2* statistics based on the rate of false alarms obtained from experiments. The SPE computed from the PCA residual was monitored using a three sigma limit approach based on a Shewhart chart. For most applications two or three principal components are sufficient to retain in a PCA model to perform the tasks of fault detection. In the experiments carried out in this work, only two principal components, $a = 2$, are retained by the PCA model. In the fault detection approach, fault alarms are generated if there is a violation of the thresholds for the *T2* statistics or the *Q* statistics, since some faults are better detected using the *T2* statistics and others by the *Q* statistics.

With respect to fault diagnosis using the influence matrix method, the performance of this method depends strongly on the multi-linearity relations between model parameters and physical parameters. Even for linear systems, the relations between model parameters and physical parameters are only approximately linear for small deviations from the nominal values, as shown by the experimental results. Assuming consistent parameter estimates, if there is a strong nonlinear relation between model parameters and physical parameters then a degradation of the performance of the fault diagnosis IMX method occurs. This is due to the fact that the influence matrix assumes that the relations are linear.

Good results have been obtained with the FDD approach based on a bank of neural recurrent output predictors (NROPs). This nonlinear FDD approach has been tested on a DC motor model, on a real DC motor and on the three tank benchmark. This FDD approach, based on input-output data, is more appropriate for additive faults, for some multiplicative faults, and for some faulty cases where other approaches normally fail, like situations of saturations, structural faults and output oscillations.

The great potential of this FDD approach comes from the fact that each neural NROP predictor contains an embedded neural model that is able to capture nonlinear dynamic behaviours. The performance of this FDD approach depends strongly on the quality of the

embedded neural output predictor models. The gain K_n of the NROP predictor is a key element since allows to guarantee the stability and to adjust the convergence properties. The design parameter, K_n, allows to settle the magnitude of the output prediction error, and consequently the fault detection and isolation performance.

The nonlinear approach to fault detection and diagnosis based on neural nonlinear principal components (neural NLPCA) combined with the pattern recognition method based on neural nonlinear discriminant analysis (neural NNLDA) reveals a good performance for the majority of faults considered. If the input-output signals are the data then the FDD approach is more appropriate to detect and diagnose additive faults. For the case of data containing ARX model parameters, the FDD approach is more suitable to deal with multiplicative (parametric) faults. In order to obtain a reasonable performance, the parameter estimation algorithm must give parameter estimates with small variances, and the patterns for each fault must be well separated in the features space. Sometimes model parameters with small variances cannot be obtained due to weak persistent excitation conditions and also to non modeled dynamics.

In all the FDD methods proposed, the fault detection and isolation signals have been low pass filtered. The low pass filters are a key element, since they allow the avoidance of false detections and wrong isolations that sometimes occur on the transients. In order to improve the reliability of FDD systems, information obtained from different methods must be combined in order to increase the overall robustness (Isermann, 1997; Palma, et al., 2002a).

The control reconfiguration problem under severe structural faults belongs to the set of complex problems in the FTC research area, since severe faults are easy to identify but difficult to correct. The experiments carried out with the three-tank benchmark showed good results. For each fault, the detection delay and the isolation delay are reasonable due to the slow dynamics of the three-tank system.

The proposed FDD methodology based on a bank of neural recurrent output predictors (NROPs) reveals a good performance to deal with severe structural faults. The fault tolerant control strategy proposed in this work for the three-tank benchmark problem showed to be effective. The adaptive optimal LQG controller performs well for the different fault scenarios, using a fixed design parameter, $r_0 = 0.03$. A better control performance is expected if the design parameter, r_0, is adjusted on-line to each faulty scenario. The dither on the reference signal has been used to guarantee persistent excitation conditions, needed for the on-line estimation of the ARX model used by the LQG controller.

6 Conclusions and Future Work

Fault tolerant systems must guarantee that faults do not cause drastic failures (L. B. Palma).

6.1 Conclusions

From a theoretical point of view, a white-box model is more desirable to perform the fault detection and diagnosis (FDD) tasks, but in most cases it is very hard, or even impossible, to obtain. When the systems are complex, or hard to model, modelling based on black-box models is usually a good, and the only, alternative.

In this dissertation, new on-line model-based fault detection and diagnosis (FDD) approaches have been proposed for linear and nonlinear dynamic systems. The FDD approaches are based on linear ARX models and nonlinear neural NARX models, using system identification techniques. In general, low order models are appropriate for fault detection and diagnosis, while high order models are more suitable for control.

One of the main ideas underlying the research carried out is the detection and diagnosis of faults in continuous time systems via the analysis of the effect on the parameters of the discrete time black-box models or related features.

The experience acquired throughout this work has shown that, in most of the situations, a combination of different FDD methodologies is necessary to implement practical and robust approaches. The model-based FDD methodologies proposed integrate different methods: analytical, data-based and knowledge-based.

Many authors argue that the parity equations and observers are most suitable for detection of abrupt additive faults, and the parameter identification is more appropriate to detect parametric faults. In fact, this statement has been confirmed by experiments in this work.

Abrupt faults have been considered in this work, but if the aim is to detect small persistent process shifts, then the cumulative sum (CUSUM) and the exponentially-weighted moving average (EWMA) charts are more appropriate than the Shewhart charts.

The neural networks of type multi-layer perceptron feed-forward (MLP-FF) have shown a great potential to perform the different tasks of output prediction, nonlinear principal components analysis and nonlinear discriminant analysis. In fact, they are a key element in this work, since they are used extensively for fault detection and isolation.

The effectiveness of the new FDD approaches has been verified by some experiments, and work well in real-time operation.

Most of the FDD approaches need an on-line estimation of ARX parameters, so they need good persistent excitation conditions and acceptable levels of noise. In this work a sliding window (SW-PCR) algorithm for parameter estimation based on principal component regression was proposed that reveals a good performance. The sliding window algorithms are more appropriate for FDD than the recursive algorithms, and this was proved based on various experiments.

The FDD approaches have been explained assuming the existence of a single fault, but they can also apply to the case of simultaneous faults. Most of the approaches described for SISO systems can be extended to MIMO systems. It is straightforward to include more faults on the FDD approaches proposed.

Many authors argue that to solve FDD problems in real applications, it is necessary to combine different FDD methodologies. The new FDD approaches proposed have been inspired by this idea and confirm this need.

The assumption of normality for the data is particularly important to compute thresholds used for fault detection and fault diagnosis. In this work, the assumption of normality for the FDD features signals, under nominal operating conditions, is a reasonable assumption. To reduce the false alarm rates in this work the fault detection and isolation signals are low pass filtered. If necessary the thresholds obtained assuming the data is normally distributed can be experimentally adjusted to guarantee a reasonable FDD performance in terms of rate of false alarms.

In all the FDD methods proposed the fault detection and isolation signals, residuals, fault alarms, etc, have been low pass filtered. The low pass filters are a key element in the FDD methods, since they allow the adjustment of the rate of false alarms, avoid false detections and wrong isolations that sometimes occur on the transients. Assuming a first order digital filter, the pole location λ must be chosen in order to obtain a desired trade-off between the rate of false alarms Ψ_{fa}, the rate of missed fault detections Ψ_{fm}, the detection delay d_d and the isolation delay d_i.

For industrial plants where the control input signals **u** are not available, and assuming that only the output signals **y** and the reference signals are available, then reference-output ARX

models $M_{yr}(\theta)$ must be used, instead of input-output ARX models $M_{yu}(\theta)$. In some industrial processes the controllers are local, and the input signal **u** is not available for the supervisor system. Since the FDD approaches proposed can use both ARX models, $M_{yu}(\theta)$ and $M_{yr}(\theta)$, they can also be applied in these situations. The ARX models $M_{yr}(\theta)$ can be used not only for fault detection and diagnosis, but also for analysis of the closed-loop dynamic performance.

The new fault detection and diagnosis methodology proposed based on dynamic features (static gain and bandwidth) of ARX models has been applied to SISO systems, and have shown good efficiency for the detection and diagnosis of multiplicative (parametric) faults. Both input-output ARX models $M_{yu}(\theta)$ and reference-output ARX models $M_{yr}(\theta)$ can be used for FDD. This approach can also detect and diagnose additive faults, and be extended to deal with nonlinear SISO systems. The neural nonlinear discriminant analysis NNLDA approach proved to be efficient for the task of pattern classification, except for the situations of fault patterns superimposed. The discriminant analysis NNLDA allows the definition of decision boundaries, in a features space of any order, and proved to be more efficient for FDI than the geometrical techniques.

The combined FDD approach, for linear systems, using principal components analysis (PCA) for fault detection and the influence matrix (IMX) method for fault diagnosis showed a reasonable performance in terms of detection and diagnosis of multiplicative (parametric) faults. It can be used also to detect additive faults. It has been formulated for SISO systems, but is straightforward to extend it to deal with MIMO systems. Both input-output ARX models $M_{yu}(\theta)$ and reference-output ARX models $M_{yr}(\theta)$ can be used for FDD. This combined FDD approach needs good persistent excitation conditions in order to obtain consistent parameter estimates.

PCA is a good technique for fault detection, but for fault diagnosis geometrical approaches or statistical approaches like discriminant analysis (Fisher FDA, NNLDA) are more appropriate. The results obtained with the PCA applied to ARX model parameters are reasonably good, taking into consideration the fact that the data does not obey a perfectly normal distribution. For most applications, two or three principal components are sufficient to retain in a PCA model to perform the tasks of fault detection. Here a three dimensional features space composed by two principal components and the SPE have been used for fault detection. Many authors argue that for most type of applications only two or three dimensional scores and the

SPE signal (based on the PCA residual) are sufficient for monitoring or FDD. The experiments carried out in this work confirm this statement.

The calculation of statistical thresholds for fault detection and isolation using PCA, based on the *T2* statistics and the three sigma limit, was made assuming that data obeys approximately a normal distribution; when necessary, a correction term was used to minimize the effects of non-perfect normality.

With respect to fault diagnosis using the influence matrix method, the performance of this method depends strongly on the multi-linearity relations between model parameters and physical parameters. Even for linear systems, the relations between model parameters and physical parameters are only approximately linear for small deviations from the nominal values, as shown by the experimental results. Assuming consistent parameter estimates, if there is a strong nonlinear relation between model parameters and physical parameters then a degradation of the performance of the fault diagnosis IMX method will occur. This is due to the fact that the influence matrix method assumes that the relations are linear.

The neural recurrent output predictor (NROP) proposed presents a good performance if the embedded neural predictor model is good enough. A good predictor model needs informative data and a good training algorithm. The Levenberg-Marquardt optimization algorithm has been used for neural network training and has shown a good performance.

Good results have been obtained with the FDD approach based on a bank of neural recurrent output predictors (NROPs). This nonlinear FDD approach, based on input-output data, is more appropriate for additive faults, for some multiplicative faults, and for some faulty cases where other approaches normally fail, like situations of saturations, structural faults and output oscillations. The square of prediction error, computed based on the NROP residual, was used for fault detection and isolation and has been shown to be a good feature. The gain K_n of the NROP predictor is a key element since it allows to guarantee the stability and to adjust the convergence properties. The design parameter, K_n, allows to settle the magnitude of the output prediction error, and consequently the fault detection and isolation performance. This approach can also deal with time-varying systems.

For detection and diagnosis of faults on nonlinear systems, the combined approach using neural nonlinear principal components (neural NLPCA) and neural nonlinear discriminant analysis (neural NNLDA) performs well for the class of faults tested. The NLPCA is applied to the parameters of ARX models estimated on-line, so this FDD method is most suitable to deal with multiplicative (parametric) faults, but it can also be used to detect some kinds of

additive faults. The neural networks in this approach play a crucial role since they are used for both the generation of features, and for the tasks of fault detection and diagnosis. The features are the nonlinear scores (in two dimensions) and the square of prediction error, and these features are used for FDD purposes using a pattern classification approach based on neural NNLDA. This approach can also deal with time-varying systems.

For control linear systems, a PI controller has been used, while for nonlinear systems an adaptive optimal linear Gaussian controller (LQGC) was used. The interaction between controller and fault detection and diagnosis is an important subject. For the FDD approaches based on system identification this is particularly important, since the FDD methods require persistent excitation conditions and these conditions must be guaranteed by the controller. Some faults cause saturations on the sensors and on the actuators and this leads to poor persistent excitation conditions. Most of the proposed FDD approaches can be applied in systems that tolerate a small control error, around 4 % of the set-point. This control error is due to the dither added to the reference signal. For the case of adaptive controllers, also based on system identification, some faulty situations cause abrupt changes on the model parameters and this can provoke situations of failures. These failures are situations where the adaptive controller degrades its performance and can no longer guarantee a small control error. To solve some of these faulty cases, it is necessary to implement strategies based on fault tolerant control.

The main advantage of Fault Tolerant Control (FTC) over other fault tolerance methods is the fact that it makes intelligent use of the redundancies included in the system, and the information about the system in order to increase the availability of the system. No method can guarantee a complete description of all possible system faults; hence no 100 % fault tolerance is possible. For many applications, only the most critical faults must be investigated and tolerated. In some cases, FTC methods cannot be sufficiently tested in operation because under practical circumstances it is usually impossible to inject some faults in the plant in order to test the reaction, so for these cases the simulation of faulty behaviours plays a very important role. Small faults are difficult to detect but easy to correct, whereas severe faults are easy to identify but difficult to correct.

In this work, the fault tolerance problem was investigated centred on the point of view of the reconfiguration problem. Reconfiguration is usually needed when severe faults occur. The control reconfiguration problem under severe structural faults, considered in this work, belongs to the set of complex problems in the FTC research area, since severe faults are easy

to identify but difficult to correct. Good results were obtained with the three-tank benchmark. For each fault, the detection delay and the isolation delay are reasonable due to the slow dynamics of the three-tank system. The FDD approach based on a bank of neural recurrent output predictors (NROPs) reveals a good performance to deal with severe structural faults. The fault tolerant control strategy proposed in this work for the three-tank benchmark problem has shown to be effective. The adaptive optimal LQG controller performs well for the different fault scenarios, using a fixed design parameter, $r_0 = 0.03$. A better control performance is expected if the design parameter, r_0, is adjusted on-line to each faulty scenario.

It is expected that the application of the new FDD methodologies proposed in this dissertation in real industrial plants will improve the overall reliability, and reduce the down-time.
Finally, it must be emphasized that in many critical fault tolerant systems the last decision belongs to the human (operator or engineer). In many cases, the supervisory systems including fault detection and diagnosis systems usually help people to take a decision about whether to shutdown a plant or not, or to take another action in a certain faulty situation.

6.2 Future Work

The combination of different approaches for fault detection and diagnosis (FDD) was particularly emphasized in this work, and is certainly one of the most promising ways for future research. One possibility is the combination of some FDD methodologies proposed, for example, FDD based on a bank of NROPs and FDD based on NLPCA-NNLDA. The combination of intelligent techniques (neural networks and fuzzy logic) for fault detection and diagnosis, and fault tolerant systems, is also a promising research area. The combination of different FDD approaches allows the detection and diagnosis of a greater class of faults.
A subject that needs more investigations is the definition of adequate thresholds, particularly for data that do not obey the normal distribution.
Abrupt faults have been considered in this work, but if the aim is to detect small persistent process shifts, then the cumulative sum and the exponentially-weighted moving average charts are more appropriate than the Shewhart charts. The detection of small persistent process shifts is also a pointer for further research.
One pointer for future research is the comparison of the sliding window algorithm SW-PCR proposed, with a fixed dimension window, and algorithms using variable forgetting factors.

The methods proposed in this dissertation can be used not only for FDD, but also to evaluate the controller performance. If reference-output ARX models, $M_{yr}(\theta)$, relating the output signal and the reference signal are used, then not only the process dynamics can be evaluated but also the controller dynamics can be examined. This is another pointer to more research in the future.

The influence matrix method assumes that linear relations between physical parameters and model parameters exist, but in practice this is not perfectly true. Future research can be done using a neural network for modeling the nonlinear relations between physical parameters and model parameters.

One formal method to select the gain of the neural recurrent output predictor (NROP) proposed must be investigated in more deep, possibly using the small gain theorem.

The area of fault tolerant control, requiring the FDD task, is a natural pointer to deep developments. The switching control is a subject that needs more research, particularly the study of the conditions to guarantee the stability of the closed-loop system. The remote network control is also an interesting area for theoretical research, and with an increasing number of industrial applications. The adaptive robust control techniques are also a promising are of research in the FTC context. There is a need for more research studies related to the interactions between system identification, control design, the fault detection and diagnosis stage and the fault tolerant control design strategies.

In future research projects, with the collaboration of other researchers and engineers from industry, there are great expectations to apply the new proposed fault detection and diagnosis approaches to industrial plants (of MIMO type) in real-time operation.

Appendix A - Terminology in the Field of Supervision, Fault Detection and Diagnosis

Some terminology given by Isermann & Ballé, based on the work within the IFAC SAFEPROCESS Technical Committee, is summarised in this Appendix, (Isermann & Balle, 1997).
By going through the literature, one recognizes immediately that the terminology in this field is not consistent.

1) STATES and SIGNALS.

Fault. An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable / usual / standard condition.

Failure. A permanent interruption of a system's ability to perform a required function under specified operating conditions.

Malfunction. An intermittent irregularity in the fulfilment of a system's desired function.

Error. A deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.

Disturbance. An unknown (and uncontrolled) input acting on a system.

Perturbation. An input acting on a system, which results in a temporary departure from the current state.

Residual. A fault indicator, based on a deviation between measurements and model-equation-based computations.

Symptom. A change of an observable quantity from normal behaviour.

2) FUNCTIONS.

Fault detection (FDE). Determination of the faults present in a system, and the time of detection.

Fault isolation (FIO). Determination of the kind, location and time of detection of a fault. Follows fault detection. The term fault classification is also used instead fault isolation.

Fault identification (FID). Determination of the size and time-variant behaviour of a fault. Follows fault isolation. The term fault analysis is also used instead of fault identification.

Fault diagnosis (FDG). Determination of the kind, size, location and time of detection of a fault. Follows fault detection. Includes fault isolation and identification.

Monitoring. A continuous real-time task of determining the conditions of a physical system, by recording information, recognising and indicating anomalies in the behaviour.

Supervision. Monitoring a physical system and taking appropriate actions to maintain the operation in the case of faults.

Protection. Means by which a potentially dangerous behaviour of the system is suppressed if possible, or means by which the consequences of a dangerous behaviour are avoided.

3) MODELS.

Quantitative model. Use of static and dynamic relations among system variables and parameters, in order to describe a system's behaviour in quantitative mathematical terms.

Qualitative model. Use of static and dynamic relations among system variables and parameters, in order to describe a system's behaviour in qualitative terms such as causalities or if-then rules.

Diagnostic model. A set of static or dynamic relations which link specific input variables - the symptoms – to specific output variables - the faults.

Analytical redundancy. Use of two or more (but not necessarily identical) ways to determine a variable, where one way uses a mathematical process model in analytical form.

4) SYSTEM PROPERTIES.

Reliability. Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time. Measure: *MTBF* = Mean Time Between Failures. $MTBF = 1 / \lambda$; λ is a rate of failure (e.g. failures per year).

Safety. Ability of a system not to cause danger to persons or equipment or the environment.

Availability. Probability that a system or equipment will operate satisfactorily and effectively at any point of time. Measure: $A = MTBF / (MTBF + MTTR)$. $MTTR = 1 / \mu$ is the Mean Time To Repair; μ is the rate of repair.

Dependability. A form of availability that has the property of always being available when required. It is the degree to which a system is operable and capable of performing its required function at any randomly chosen time during its specified operating time, provided that the item is available at the start of that period. The dependability is expressed by $D = Time\ available / (Time\ available + Time\ required)$.

Other terminology in this field can also be found in the papers (Venkatasubramanian, et al., 2003a & 2003b & 2003c). Terminology more related to Maintenance and Reliability can be found in the book (Assis, 2004).

References

Ackoff, R. (1989), From Data to Wisdom, *Journal of Applied Systems Analysis*, vol. 16, pp. 3-9.

Antory, D., U. Kruger, G. Irwin, G. McCullough (2004), Industrial Process Monitoring Using Nonlinear Principal Component Models, *Proc. of 2nd IEEE Conf. of Intelligent Systems*, Bulgaria.

Antory, D., U. Kruger, G. Irwin, G. McCullough (2005), Fault Diagnosis in Internal Combustion Engines Using Nonlinear Multivariate Statistics, *Journal of Systems & Control Eng.*, vol. 219, no. 4, pp. 243-258.

Asoh, H., N. Otsu (1990), An Approximation of Nonlinear Discriminant Analysis by Multilayer Neural Networks, *IEEE Int. Joint Conf. on Neural Networks* (IJCNN), San Diego, USA.

Assis, R. (2004), *Apoio à Decisão em Gestão da Manutenção*, Lidel, Portugal.

Astrom, K., B. Wittenmark (1995), *Adaptive Control*, Addison-Wesley.

Astrom, K., B. Wittenmark (1997), *Computer-Controlled Systems*, Prentice-Hall.

Astrom, K., T. Hagglund (1988), *Automatic Tuning of PID Controllers*, Instrument Society of America.

Astrom, K., T. Hagglund (2000a), The Future of PID Control, *Proc. of the IFAC Workshop on Digital Control – Past, Present and Future of PID Control,* April 5-7, Terrassa – Spain.

Astrom, K., T. Hagglund (2000b), Benchmark Systems for PID Control, *Proc. of the IFAC Workshop on Digital Control – Past, Present and Future of PID Control,* April 5-7, Terrassa – Spain.

Basseville, M. (1988), Detecting Changes in Signals and Systems – A Survey, *Automatica*, vol. 24, pp. 309-326.

Basseville, M., I. Nikiforov (1993), *Detection of Abrupt Changes: Theory and Application*, Prentice-Hall.

Beard, R. (1971), Failure Accommodation in Linear Systems through Self-Reorganization, Phd Thesis, M.I.T., Man Vehicle Lab, Cambridge, Massachusetts.

Berry, M., G. Linoff (1997), *Data Mining Techniques*, Wiley.

Bishop, C. (1995), *Neural Networks for Pattern Recognition*, Clarendon Press, New York.

Blanke, M., M. Kinnaert, J. Lunze, M. Staroswiecki (2003), *Diagnosis and Fault-Tolerant Control*, Springer.

Chen, J., R. Patton (1999), *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers.

Chiang, L., E. Russell, R. Braatz (2001), *Fault Detection and Diagnosis in Industrial Systems*, Springer-Verlag.

Chow, E., A. Willsky (1984), Analytical Redundancy and the Design of Robust Failure Detection Systems, *IEEE Trans. on Automatic Control*, vol. 29, no. 7, pp. 603-614.

Conover, W. (1999), *Practical Nonparametric Statistics*, Wiley.

Diamantaras, K. (1996), *Principal Component Neural Networks – Theory and Applications*, John Wiley & Sons.

Ding, X., P. Frank (1990), Fault Detection via Factorization Approach, *Systems Control Letter*, vol. 14, pp. 431-436.

Duda, R., P. Hart, D. Stork (2000), *Pattern Classification*, 2nd ed., Wiley.

Doraiswami, R., M. Stevenson (1996), A Robust Influence Matrix Approach to Fault Diagnosis, *IEEE Transactions on Control Systems Technology*, vol. 4, no. 1, pp. 29-39.

Driankov, D., H. Hellendoorn, M. Reinfrank (1996), *An Introduction to Fuzzy Control*, Springer.

Dunia, R., S. Qin, T. Edgar (1995), Multivariable Process Monitoring using Nonlinear Approaches, *Proc. of the American Control Conference*, pp. 756-760, Seattle, IEEE Press.

Farinwata, S., D. Filev, R. Langari (2000), *Fuzzy Control – Synthesis and Analysis*, John Wiley & Sons.

Farrell, J. (1996), Neural Control - Analysis and Design of Nonlinear Systems, *The Control Handbook*, edited by W. Levine, pp. 1017-1030, CRC Press, USA.

Filev, D. (2000), Gain Scheduling Based Control of a Class of TSK Systems, *Fuzzy Control*, edited by S. Farinwata, D. Filev, R. Langari, John Wiley & Sons.

Fisher, R. (1936), The Use of Multiple Measurements in Taxonomic Problems, *Ann. Eugenics,* vol. 7, pp. 179-188.

Frank, P. (1990), Fault Diagnosis in Dynamic Systems using Analytical and Knowledge-Based Redundancy – A Survey and Some New Results, *Automatica*, vol. 26, no. 3.

Frank, P. (1996), Analytical and Qualitative Model-Based Fault Diagnosis – A Survey and Some New Results, *European Journal of Control*, vol. 2, pp. 6-28.

Frank, P., E. Garcia, B. Koppen-Seliger (1997), Modelling for Fault Detection and Isolation, *Proc. of the 3rd Cosy Workshop on Control of Complex Systems*, Budapest, Hungary.

Frank, P., S. Ding, T. Marcu (2000a), Model Based Fault Diagnosis in Technical Processes, *Trans. of the Inst. of Measurement and Control*, vol. 22, no. 1.

Frank, P., S. Ding, B. Koppen-Seliger (2000b), Current Developments in the Theory of FDI, *Proc. of Safeprocess Symposium*, Budapest-Hungary.

Franklin, G., J. Powell, M. Workman (1998), *Digital Control of Dynamic Systems*, Addison-Wesley.

Friedland, B. (1996), Observers, *The Control Handbook*, edited by W. Levine, pp. 607-618, CRC Press, USA.

Gallinari, P., S. Thiria, F. Badran, F. Fogelman-Soulie (1991), On the Relations between Discriminant Analysis and Multilayer Perceptrons, *Neural Networks*, vol. 4, pp. 349-360.

Geladi, P., B. Kowalsky (1986), Partial Least Squares Regression: A Tutorial, *Analytica Chimica Acta*, vol. 185, pp. 1-17.

Genrup, M. (2005), *On Degradation and Monitoring Tools for Gas and Steam Turbines*, Phd Thesis, Lund University, Sweden.

Gertler, J. (1988), Survey of Model-Based Failure Detection and Isolation in Complex Plants, *IEEE Control Systems Magazine*, vol. 8, no. 6.

Gertler, J., (1998), *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker Inc.

Gertler, J. (2000), All Linear Methods Are Equal – And Extendible To Nonlinearities, *Proc. of Safeprocess Symposium*, Budapest-Hungary.

Gnanadesikan, R. (1997), *Methods for Statistical Data Analysis of Multivariate Observations*, John Wiley & Sons.

Golub, G., C. Loan (1996), *Matrix Computations*, Johns Hopkins University Press.

Gower, J., D. Hand (1996), *Biplots*, Chapman & Hall, London.

Gustafsson, F., (2001), *Adaptive Filtering and Change Detection*, John Wiley & Sons.

Hagan, M., H. Demuth, M. Beale (1995), *Neural Network Design*, PWS Publishing Company, Boston - USA.

Harkat, M. (2003), *Détection et Localisation de Défauts par Analyse en Composantes Principales*, PHD Thesis, Institut National Polytechnique de Lorraine, France.

Hastie, T. (1984), *Principal Curves and Surfaces*, Phd Thesis, Stanford University, USA.

Haykin, S. (1994), *Neural Networks – A Comprehensive Foundation*, MacMillan College Publishing Company, Inc.

Haykin, S. (2002), *Adaptive Filter Theory*, Prentice-Hall.

Heiming, B., J. Lunze (1999), Definition of the Three-Tank Benchmark Problem for Controller Reconfiguration, *Proc. of the European Control Conference ECC'99*, Karlsruhe, Germany.

Himes, D., R. Storer, C. Georgakis (1994), Determination of the Number of Principal Components for Disturbance Detection and Isolation, *Proc. of the American Control Conference*, pp. 1279-1283, New Jersey, IEEE Press.

Hippenstiel, R. (2002), *Detection Theory – Applications and Digital Signal Processing*, CRC Press.

Hsieh, W. (2001), Nonlinear Principal Component Analysis by Neural Networks, *Tellus Journal*, vol. 53A, pp. 599-615.

Isermann, R. (1984), Process Fault Detection based on Modeling and Estimation Methods, *Automatica*, vol. 20, pp. 387-404.

Isermann, R. (1994), Integration of Fault Detection and Diagnosis Methods, *Proc. of IFAC Safeprocess Symposium*, Espoo - Finland.

Isermann, R. (1995), Model Based Fault Detection and Diagnosis Methods, *Proc. of the American Control Conference*, Seattle, USA.

Isermann, R. (1997), Supervision, Fault Detection and Fault Diagnosis Methods – An Introduction, *Control Eng. Practice*, vol. 5, no. 5, pp. 639-652.

Isermann, R., P. Balle (1997), Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes, *Control Eng. Practice*, vol. 5, no. 5, pp. 709-719.

Isermann, R. (2004), Model-Based Fault Detection and Diagnosis – Status and Applications, *Proc. of the IFAC Symposium on Automatic Control in Aerospace*, June 14-18, St. Petersburg, Russia.

Jackson, J. (2003), *A User's Guide To Principal Components*, Wiley.

Jackson, J., G. Mudholkar (1979), Control Procedures for Residuals Associated with Principal Component Analysis, *Technometrics*, vol. 21, pp. 341-349.

Johansson, R. (1993), *System Modeling and Identification*, Prentice Hall.

Jolliffe, I. (2002), *Principal Component Analysis*, Springer.

Jones, H. (1973), *Failure Detection in Linear Systems*, Phd Thesis, Dept. of Aeronautics, M.I.T., Cambridge, Massachusetts.

Jussila, T. (1992), *Computational Issues of PID Controllers*, Phd Thesis, Tampere University of Technology, Finland.

Kalman, R. (1960), A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME*, vol. 82, pp. 35-45.

Kesavan, P., J. Lee (1997), Diagnostic Tools for Multivariable Model-Based Control Systems, *Ind. Eng. Chem. Res.*, vol. 36, pp. 2725-2738.

Klancar, G. (2000), Fault Detection and Isolation by Means of Principal Component Analysis, *Cybernetics & Informatics Eurodays Workshop*, Sept. 26-30, Prague, Czech Republic.

Koppen-Seliger, B., P. Frank (1995), Fault Detection and Isolation in Technical Processes with Neural Networks, *Proc. of the IEEE 34th Conference on Decision and Control*, New Orleans, USA.

Koppen-Seliger, B., E. Garcia, P. Frank (1999), Fault Detection: Different Strategies for Modelling Applied to the Three Tank Benchmark – A Case Study, *Proc. of the European Control Conference ECC'99*, Karlsruhe, Germany.

Kosko, B. (1994), Fuzzy Systems as Universal Approximators, *IEEE Trans. on Computers*, vol. 43, pp. 1329-1333.

Kramer, M. (1991), Nonlinear Principal Component Analysis using Auto-Associative Neural Networks, *AICHE Journal*, vol. 37, no. 2, pp. 233-243.

Kuo, B. (1995), *Automatic Control Systems*, Prentice-Hall.

Leonhardt, S., M. Ayoubi (1997), Methods of Fault Diagnosis, *Control Eng. Practice*, vol. 5, no. 5, pp. 683-692.

Lewis, F. (1996), Optimal Control, *The Control Handbook*, edited by W. Levine, CRC Press.

Liberzon, D. (2003), *Switching in Systems and Control*, Birkhauser.

Lilliefors, H. (1967), On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown, *Journal of the American Statistical Association*, vol. 62.

Ljung, L. (1999), *System Identification – Theory for the User*, Prentice Hall.

Lopes, J. (2001), *Supervisão e Diagnóstico de Processos Farmacêuticos com Métodos Inteligentes de Análise de Dados*, PHD Thesis, Universidade Técnica de Lisboa - IST, Lisboa – Portugal.

Luenberger, D. (1966), Observers for Multivariate Systems, *IEEE Trans. on Autom. Control*, vol. AC-11, no. 2, pp. 190-197.

Lutkepohl, H. (1996), *Handbook of Matrices*, John Wiley & Sons.

Man, K., K. Tang, S. Kwong (1999), *Genetic Algorithms*, Springer.

Marcu, T., L. Mirea, P. Frank (1998), Neural Observer Schemes for Robust Detection and Isolation of Process Faults, *Proc. of IEE UKACC International Conference on Control*.

Marcu, T., M. Matcovschi, P. Frank (1999), Neural Observer-Based Approach to Fault Detection and Isolation of a Three-Tank System, *Proc. of the European Control Conference ECC'99*, Karlsruhe, Germany.

Marques de Sá, J. (2001), *Pattern Recognition – Concepts, Methods and Applications*, Springer.

Martinez, W., A. Martinez (2002), *Computational Statistics Handbook with Matlab*, Chapman and Hall / CRC Press.

Montgomery, D. (1991), *Introduction to Statistical Quality Control*, Wiley.

Mosca, E. (1995), *Optimal, Predictive, and Adaptive Control*, Prentice Hall.

Moseler, O., M. Muller (2000), A Smart Actuator with Model-Based FDI Implemented on a Microcontroller, *1st IFAC Conference on Mechatronic Systems*, Sept. 18-20, Darmstadt, Germany.

Narendra, K., K. Parthasarathy (1990), Identification and Control of Dynamical Systems using Neural Networks, *IEEE Trans. on Neural Networks*, vol. 1, no.1, pp. 4-27.

Norgaard, M., O. Ravn, N. Poulsen, L. Hansen (2003), *Neural Networks for Modelling and Control of Dynamic Systems*, Springer.

Ogunnaike, B., W. Ray (1994), *Process Dynamics, Modeling, and Control*, Oxford University Press, New York.

Ono, T., T. Kumamaru, A. Maeda, S. Sagara, K. Kumamaru (1987), Influence Matrix Approach to Fault Diagnosis of Parameters in Dynamic Systems, *IEEE Transactions on Industrial Electronics*, vol. IE34, no. 2, pp. 285-291.

Oppenheim, A., A. Willsky, I. Young (1983), *Signals and Systems*, Prentice-Hall.

Palma, L. Brito, R. Neves da Silva, F. Vieira Coito (2002a), Metodologia Híbrida de Detecção e Diagnóstico de Falhas em Tempo Real, *Proc. of Conferência Científica e Tecnológica em Engenharia*, CCTE-2002, Maio 6-10, ISEL, Lisboa-Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2002b), Adaptive Observer Based Fault Diagnosis Approach Applied to a Thermal Plant, *Proc. of the 10th IEEE Mediterranean Conference on Control and Automation*, MED2002, July 9-12, UTL-IST, Lisboa-Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2002c), On-Line Fault Diagnosis Integrated Approach: Application to a Thermal Process, *Proc. of the 5th Portuguese Conference on Automatic Control*, Controlo 2002, Sept. 5-7, Univ. of Aveiro, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2003a), Neural Observer-Based Approach to Fault Diagnosis Applied to a Liquid Level System, *Proc. of the IFAC International Conference on Intelligent Control Systems and Signal Processing*, ICONS 2003, Apr. 8-11, Univ. of Algarve, Portugal.

Palma, L. Brito, R. Neves da Silva, F. Vieira Coito (2003b), Fault Tolerant Control Approach Applied to the Three-Tank System, *Proc. of the 8th Portuguese-Spanish Congress in Electrical Engineering*, 8CLEEE, July 3-5, Vilamoura, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2003c), Fault Diagnosis based on Black-Box Models with Application to a Liquid-Level System, *Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation,* ETFA'03, Sept. 16-19, Lisbon, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2004a), Sensor Fault Diagnosis based on Neural Observers and Parameter Estimation – Application to the Three-Tank Benchmark, *Proc. of the 6th Portuguese Conference on Automatic Control*, Controlo'04, University of Algarve, June 7-9, Faro, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2004b), Robust Fault Diagnosis Approach using Analytical and Knowledge-Based Techniques Applied to a Water Tank System, *Proc. of the International Conference on Knowledge Engineering and Decision Support*, ICKEDS'04, July 21-23, Porto, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2004c), A Combined Approach to Fault Diagnosis in Dynamic Systems – Application to the Three-Tank Benchmark, *Proc. of the IEEE International Conference on Informatics in Control, Automation, and Robotics*, ICINCO'04, Polytechnic Institut of Setúbal, Aug. 25-28, Setúbal, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2005a), A Combined Approach to Fault Diagnosis based on Principal Components and Influence Matrix, *Proc. of the IEEE International Symposium on Intelligent Signal Processing*, WISP'05, University of Algarve, Sept. 1-3, Faro, Portugal.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2005b), Process Fault Diagnosis Approach based on Neural Observers, *Proc. of the 10th IEEE International Conference on Emerging Technologies and Factory Automation*, ETFA'05, Facoltá di Ingegneria, Sept. 19-22, Catania, Italy.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2005c), Diagnosis of Parametric Faults based on Identification and Statistical Methods, *Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference*, CDC-ECC'05, Dec. 12-15, Seville, Spain.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva (2005d), Robust Fault Diagnosis Approach using Analytical and Knowledge-Based Techniques Applied to a Water Tank System, *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications – Special Issue on Knowledge Engineering*, CRL Publishing Ltd, vol. 13, no. 4, pp. 237-244.

Palma, L. Brito, F. Vieira Coito, R. Neves da Silva, R. Filipe Almeida (2006), A Neural PCA Approach To Fault Detection and Diagnosis in Nonlinear Dynamical Systems, *International Conference on Knowledge Engineering and Decision Support (ICKEDS'06)*, May 9-12, Lisboa, Portugal.

Patton, R. (1997), Fault Tolerant Control Systems: the 1997 Situation, *Proc. of the Safeprocess Symposium*, Hull, UK.

Patton, R., F. Uppal, C. Toribio (2000), Soft Computing Approaches to Fault Diagnosis for Dynamic Systems: A Survey, *Proc. of the IFAC Safeprocess Symposium*, Budapest-Hungary.

Pau, L. (1981), *Failure Diagnosis and Performance Monitoring*, Marcel Dekker, New York.

Piovoso, M., K. Kosanovich (1994), Applications of Multivariate Statistical Methods to Process Monitoring and Controller Design, *Int. J. of Control*, vol. 59, pp. 743-765.

Piovoso, M., K. Kosanovich (1996), The Use of Multivariate Statistics in Process Control, *The Control Handbook* , edited by W. Levine, pp. 561-573, CRC Press, USA.

Poshtan, J., R. Doraiswami, and M. Stevenson (1997), A Real-Time Fault Diagnosis and Parameter Tracking Scheme, *Proc. of the American Control Conference*, New Mexico, Albuquerque, pp. 483-487.

Poshtan, J., R. Doraiswami (1994), Influence Matrix Approach to Fault Diagnosis and Controller Tuning, *IEEE Canadian Conference on Electrical and Computer Engineering*, Canada.

Raich, A., A. Cinar (1996), *Statistical Process Monitoring and Disturbance Diagnosis in Multivariate Continuous Processes*, AICHE Journal, no. 42, pp. 995-1009.

Ramos, J. (1998), *Projecto e Aplicação de Controladores Difusos em Tempo-Real*, Msc Thesis, Univ. de Coimbra, Portugal.

Salvador Marques, J. (1999), *Reconhecimento de Padrões: Métodos Estatísticos e Neuronais*, IST Press, Lisboa, Portugal.

Santina, M., A. Stubberud, G. Hostetter (1996), Sample-Rate Selection, *The Control Handbook*, edited by W. Levine, CRC Press.

Setnes, M., R. Babuska (2000), Fuzzy Modeling for Predictive Control, in *Fuzzy Control*, edited by S. Farinwata, D. Filev, R. Langari, John Wiley & Sons.

Sjoberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Glorennec, H. Hjalmarsson, A. Judistsky (1995), Nonlinear Black-Box Modeling in System Identification: A Unified Overview, *Automatica*, vol. 31, pp. 1691-1724.

Soderstrom, T., P. Stoica (1989), *System Identification*, Prentice-Hall.

Takagi, T., M. Sugeno (1985), Fuzzy Identification of Systems and its Application to Modeling and Control, *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15, pp. 116-132.

Venkatasubramanian, V., R. Rengaswamy, K. Yin, S. Kavuri (2003a), A Review of Process Fault Detection and Diagnosis. Part I: Quantitative Model-Based Methods, *Computers and Chemical Engineering*, vol. 27, pp. 293-311.

Venkatasubramanian, V., R. Rengaswamy, S. Kavuri (2003b), A Review of Process Fault Detection and Diagnosis. Part II: Qualitative Models and Search Strategies, *Computers and Chemical Engineering*, vol. 27, pp. 313-326.

Venkatasubramanian, V., R. Rengaswamy, S. Kavuri, K. Yin (2003c), A Review of Process Fault Detection and Diagnosis. Part III: Process History Based Methods, *Computers and Chemical Engineering*, vol. 27, pp. 327-346.

Wang, X. (1999), *Data Mining and Knowledge Discovery for Process Monitoring and Control*, Springer-Verlag, London.

Wang, X., U. Kruger, G. Irwin (2005), Process Monitoring Approach Using Fast Moving Window PCA, *Ind. Eng. Chem. Res.*, vol. 44, pp. 5691-5702.

Wei, W. (1990), *Time Series Analysis*, Addison-Wesley.

Welch, G., G. Bishop (2001), An Introduction to the Kalman filter, internal paper, Department of Computer Science, University of North Carolina.

Wise, B, N. Ricker, D. Veltkamp (1989), Upset and Sensor Failure Detection in Multivariate Processes, technical report, Eigenvector Research, Manson, Washington.

Wise, B., N. Ricker (1990), The Effect of Biased Regression on the Identification of FIR and ARX Models, *Proc. of the AIChe Annual Meeting*, Chicago – USA.

Wise, B. (1991), *Adapting Multivariate Analysis for Monitoring and Modeling of Dynamic Systems*, Phd Thesis, Dept. of Chem. Eng., Univ. of Washington, USA.

Wise, B., N. Gallagher (1996), The Process Chemometrics Approach to Process Monitoring and Fault Detection, *Journal of Process Control*, vol. 6, no. 6, pp. 329-348.

Yager, R., D. Filev (1994), *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons.

Zar, J. (1999), *Biostatistical Analysis*, Prentice-Hall.

Zhang, J., E. Martin, A. Morris (1995), Fault Detection and Classification through Multivariate Statistical Techniques, *Proc. of the American Control Conference*, pp. 751-755, New Jersey, IEEE Press.

Zhou, J., S. Bennett (1997), Dynamic System Fault Diagnosis based on Neural Network Modelling, *Proc. of the IFAC Safeprocess Symposium*, Univ. of Hull, UK.